

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra kybernetiky a biomedicínského inženýrství

**Automatizována detekce a třídění typů tablet pomocí obrazového
signálu**
**Autonomous Detection and Separation of Pill Types from Camera
Signal**

Zadání bakalářské práce

Student:

Ivana Večerková

Studijní program:

B2649 Elektrotechnika

Studijní obor:

3901R039 Biomedicínský technik

Téma:

Automatizována detekce a třídění typů tablet pomocí obrazového signálu

Autonomous Detection and Separation of Pill Types
from Camera Signal

Zásady pro vypracování:

1. Rozbor problematiky analýzy obrazového signálu a metod detekce s využitím poznatků biomedicíny.
2. Návrh metod pro analýzu tvarů, velikosti a barev z obrazového signálu.
3. Realizace metod analýzy a detekčního systému s navrženými algoritmy.
4. Srovnání naměřených a analyzovaných výsledků s teoretickými předpoklady.
5. Zhodnocení dosažených výsledků.

Seznam doporučené odborné literatury:

- [1] LÜLLMANN, Heinz, Klaus MOHR and Martin WEHLING. *Farmakologie a toxikologie: překlad 15., zcela přepracovaného vydání*. Vyd. 2. české. Praha: Grada Publishing, 2004. 725 s. ISBN 80-247-0836-1.
- [2] ELLER, Frank. *C# : začínáme programovat : podrobný průvodce začínajícího uživatele*. 1.vyd. Praha: Grada Publishing, 2002. 240 s. ISBN 80-247-0324-6.
- [3] DOBEŠ, Michal. *Zpracování obrazu a algoritmy v C#*. 1.vyd. Praha: BEN - technická literatura, 2008. 143 s. ISBN 978-80-7300-233-6.
- [4] HLAVÁČ, Václav a Miloš SEDLÁČEK. *Zpracování signálů a obrazů*. 2.přeprac.vyd. Praha: Nakladatelství ČVUT, 2007. 255 s. ISBN 978-80-01-03110-0.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Zdeněk Macháček, Ph.D.**

Datum zadání: 16.11.2012

Datum odevzdání: 07.05.2014

doc. Ing. Jiří Koziorek, Ph.D.
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

„Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně. Uvedla jsem všechny literární prameny a publikace, ze kterých jsem čerpala.“

V Ostravě 7. 5. 2014

.....*Květa Kovář*.....

Ráda bych poděkovala vedoucímu bakalářské práce Ing. Zdeňkovi Macháčkovi Ph.D. za odborné vedení, cenné rady a připomínky. Dále bych ráda poděkovala svému bratranci Jiřímu Kupkovi za rady při psaní algoritmů. Současně děkuji svým rodičům a příteli, kteří mě po celou dobu studia plně podporovali.

Abstrakt

Tato bakalářská práce byla zpracována na téma automatizovaná detekce a třídění typů tablet pomocí obrazového signálu. Detekce a třídění je prováděná pomocí NXT kamery a barevného světelného senzoru. Mnou navržená třídící linka byla sestavena pomocí stavebnice LEGO MINDSTORMS, jejíž základní součástí je řídicí jednotka LEGO NXT (řídicí kostka NXT). Řídicí jednotka disponuje 32 bitovým mikroprocesorem a 4 vstupními a 3 výstupními porty. Programování třídící linky proběhlo v programu RobotC. Programování bylo prováděno textovým zápisem příkazu jazyka C v kombinaci s příkazy získávající informace ze senzorů a ovládající výstupní zařízení. Při analýzách obrazu jsou využívány knihovny dodávané výrobcem. Třídící linka je navržena jako systém posuvných pásu, na kterých se nejprve detekuje pomocí kamery tvar a velikost projíždějící tablety a pak pomocí barevného světelného senzoru barva tablety. Následně je rozlišená tableta umístěna do příslušné části zásobníku.

Klíčová slova

LEGO, řídicí jednotka NXT, barevný světelný senzor, NXT kamera

Abstract

This bachelor thesis was elaborated on the topic of Autonomous Detection and Separation of Pill Types from Camera Signal. Detection and sorting is performed using a camera and a light color sensor. My proposed sorting line was built using LEGO MINDSTORMS, the basic component is the control unit LEGO NXT (NXT control brick). The control unit has a 32-bit microprocessor and 4 input and 3 output ports. Programming sorting line was in the program to the RobotC. Programming was done by writing a text command language C in combination with the dictates of receiving information from sensors and controlling output devices. For analyzes the image are used libraries supplied by programmers. The sorting line is designed as a system of sliding of the belt on which is first detected by the camera size and shape of the tablets, and then passing using a colored light sensor color tablets. Subsequently, the resolution tablet is placed in the appropriate part of the tank.

Keywords

LEGO, NXT the control unit, color light sensor, the camera

Seznam použitých symbolů a zkratk

CCD	(Charge Coupled Device) technologie pro obrazový senzor
CIE	(Commission International de l'Éclairage) první matematicky definovaný barevný prostor
CMOS	(Complementary metal-oxide-semiconductor) technologie pro obrazový senzor
CMY	(Cyan, Magenta, Yellow) barevný model
CMYK	(Cyan, Magenta, Yellow, black) barevný model
Hz	(Hertz) jednotka frekvence (kmitůtu)
EV3	(Evolution - 3) evoluce 3 generace
LED	(Light-Emitting-Diode) dioda emitující světlo
LEGO	(leg godt) dobrá hra
PC	(personal computer) osobní počítač
RGB	(Red, Green, Blue) barevný model
SD	(secure digital) paměťová karta
USB	(universal serial bus) – univerzální sériová sběrnice
V	volt
XP	(eXPerience) zkušenost
bit	(binary digit) dvojková číslice, =drobek, kousek
cm	centimetr
mA	miliampér
mm	milimetr
nm	nanometr
n. l.	našeho letopočtu
př. n. l.	před naším letopočtem
px	(pixel- picture element) obrazový prvek
s	sekunda
wifi	(Wireless fidelity) tzv. bezdrátová věrnost

OBSAH

ÚVOD.....	1
1. ROZBOR PROBLEMATIKY ANALÝZY OBRAZOVÉHO SIGNÁLU A METOD DETEKCE S VYUŽITÍM POZNATKŮ BIOMEDICÍNY	2
1.1 ROZBOR MEDICÍNSKÝCH POZNATKŮ A ROZDĚLENÍ TYPŮ LÉČIV	2
1.1.1 Historie léčiv a léků.....	2
1.1.2 Pojem lék a léčivo	2
1.1.3 Lékové formy	3
1.1.3.1 Pevné lékové formy.....	3
1.1.3.2 Polotuhé lékové formy	5
1.1.3.3 Kapalné lékové formy	5
1.2 ANALÝZA POUŽITÝCH VÝVOJOVÝCH PROSTŘEDKŮ	7
1.2.1 Historie LEGO	7
1.2.2 Popis modulů NXT.....	8
1.2.2.1 Řídící jednotka NXT (Řídící kostka NXT)	8
1.2.2.2 Spojovací kabely NXT	10
1.2.2.3 Senzory.....	10
1.2.2.3.1 Světelný senzor	10
1.2.2.3.2 Dotykový senzor	11
1.2.2.3.3 Barevný světelný senzor.....	12
1.2.2.4 Interaktivní servomotory	12
1.2.2.5 NXT kamera.....	13
1.3 ANALÝZA ZPRACOVÁNÍ DIGITÁLNÍHO OBRAZU	14
1.3.1 Digitalizace.....	14
1.3.2 Barevné modely.....	14
1.3.2.1 Barevný model RGB	14
1.3.2.2 Barevný model CMY a CMYK.....	15
2. NÁVRH METOD PRO ANALÝZU TVARU, VELIKOSTI A BAREV Z OBRAZOVÉHO SIGNÁLU	16
2.1 PROGRAMOVÁNÍ TŘÍDÍCÍ LINKY	16
2.1.1 Matlab	16
2.1.2 NXC	17
2.1.3 RobotC	17
2.1.3.1 Postup nahrání firmware	18
2.1.4 NXTCamView	19

3. NÁVRH A REALIZACE ŘEŠENÍ.....	21
3.1 VYTVOŘENÍ TŘÍDÍCÍHO MECHANISMU S POUŽITÍM INTERAKTIVNÍCH SERVOMOTORŮ.....	24
3.1.1 Kód použitý v programu RobotC pro motor posuvného pásu A.....	24
3.1.2 Kód použitý v programu RobotC pro motor posuvného pásu A a B	24
3.2 VYTVOŘENÍ ROZLIŠOVACÍHO MECHANISMU S VYUŽITÍM BAREVNÉHO SVĚTELNÉHO SENZORU.....	25
3.2.1 Načtení knihovny pro barevný světelný senzor.....	26
3.2.2 Kód použitý v programu RobotC pro načtení barvy	26
3.2.3 Kód použitý v programu RobotC pro zařazení barvy v posuvném zásobníku ...	26
3.3 VYTVOŘENÍ ROZLIŠOVACÍHO MECHANISMU S VYUŽITÍM NXT KAMERY	27
3.3.1 Načtení knihovny pro NXT kameru	27
3.3.2 Kód pro určení tvaru v programu RobotC.....	28
3.3.3 Kód pro určení velikosti a zařazení pro kulatou tabletu v programu RobotC	28
4. VERIFIKACE NAMĚŘENÝCH A ANALYZOVANÝCH VÝSLEDKŮ	29
4.1 ÚSPĚŠNOST ZAŘAZENÍ TABLET	30
4.2 ČETNOST CHYB PŘI DETEKCI OBOU SENZORŮ	31
4.3 ČETNOST CHYB PŘI DETEKCI NXT KAMERY	32
4.4 ČETNOST CHYB PŘI DETEKCI BAREVNÉHO SVĚTELNÉHO SENZORU	33
4.5 SROVNÁNÍ ČETNOSTI CHYB U DETEKCE NXT KAMERY A BAREVNÉHO SVĚTELNÉHO SENZORU.....	34
4.6 ZDŮVODNĚNÍ SPRÁVNÉHO A ŠPATNÉHO ZAŘAZENÍ TABLETY U BAREVNÉHO SVĚTELNÉHO SENZORU.....	35
4.7 ZDŮVODNĚNÍ SPRÁVNÉHO A ŠPATNÉHO ZAŘAZENÍ TABLETY U NXT KAMERY	36
ZÁVĚR.....	37
SEZNAM POUŽITÉ LITERATURY	38
INTERNETOVÉ ZDROJE	38
BAKALÁŘSKÁ PRÁCE.....	38
SEZNAM OBRÁZKŮ	39
SEZNAM TABULEK	40
SEZNAM GRAFŮ	40
SEZNAM POUŽITÝCH PROGRAMŮ.....	41
SEZNAM PŘÍLOH.....	42

Úvod

Tato bakalářská práce je zaměřená na automatizovanou detekci a třídění typu tablet pomocí obrazového signálu.

Výroba léčiv, léků, veterinárních preparátů, potravinových doplňků a dalších se v dnešní době neustále vyvíjí a rozšiřuje. Zavedení detekce a následného třídění pomocí obrazového signálu by mohlo být využito jako součást farmaceutické linky pro výrobu léčiv, a to nejen tablet, ale taky tobolek a jiných pevných lékových forem.

Bakalářská práce popisuje způsob programování jednotlivých částí přístroje LEGO MINDSTORMS, který dále využiji pro detekci a následné třídění tablet. Tablety zařazujeme do kategorie pevné lékové formy. Mohou být různého tvaru, velikosti, povrchové struktury a barev. Mezi další pevné lékové formy můžeme uvést tobolky, kapsle, dražé, granuláty, čípky atd.

Návrh třídící linky je sestaven s komponentů stavebnice LEGO MINDSTORMS. Základní řídicí jednotka NXT je LEGO kostka, (řídicí jednotka NXT) na kterou je možné díky vstupům a výstupům připojit další příslušenství stavebnice (interaktivní servomotor, senzory atd.)

V třídící lince je použita NXT kamera o rozlišení 88 x 144 pixelů. Slouží pro detekci tvaru a velikosti tablet. Barevný světelný senzor slouží pro rozlišení barvy tablet. Po rozlišení jsou tablety zařazeny do zásobníku.

Třídící linka bude podrobena 20 sérií pokusů o 16 tabletách. Použité tvary jsou kulaté a podlouhlé, velikosti velké a malé. Barvy jsou žluté, červené a růžové. Z provedených pokusů bude zaznamenán do tabulky počet správně a špatně zařazených tablet. Dále počty chybně zařazených tablet u obou detekčních zařízení dohromady (NXT kamera a barevný světelný senzor) a u každého ze senzorů zvlášť. Pro pokusy byly použity místo tablet pevné cukrovinky.

Možné využití detekce by mohlo být v různých odvětvích. Jako např. kontrola přilepení značek, popisků a etiket, kontrola správného obsahu láhví, počítání různých dílů a součástek ve strojírenském průmyslu a mnohé další.

1. ROZBOR PROBLEMATIKY ANALÝZY OBRAZOVÉHO SIGNÁLU A METOD DETEKCE S VYUŽITÍM POZNATKŮ BIOMEDICÍNY

1.1 Rozbor medicínských poznatků a rozdělení typů léčiv

1.1.1 Historie léčiv a léků

Působení některých rostlinných látek na lidský organismus bylo známo již ve starověku. Již 3 tisíce let př. n. l. znali Babyloňané opium a další drogy. V Ebersově papýru se dovídáme recepty starých Egyptů, jak připravovali různé léky. Ze starověkých lékařů je třeba se zvláště zmínit o Galénovi (129 – 201 n. l.), který zobecnil metody zpracování různých léčiv. Ve své léčitelské praxi používal léky složené, které si sám připravoval. Podle něj byla pojmenována věda o technologii léků = galenika. Teprve novověk přinesl s rozvojem medicíny i rozvoj farmaceutických věd. [1]

1.1.2 Pojem lék a léčivo

Chalabala et al. uvádí jako léčivo látku nebo soustavu látek, která má schopnost interagovat s lidským nebo živočišným organismem a výsledkem této interakce je poznání chorob, ochrana před nimi, mírnění příznaků onemocnění a jejich léčení (na některé z těchto funkcí se léčiva záměrně vyhledávají, účelně a cílevědomě používají). Léčiva nejsou vhodná pro přímé použití, proto je potřeba upravit je tak aby mohla na daném místě, v daném čase plnit požadovanou funkci, transformovat je do podoby léků.

Lék je produkt získaný z léčiv a farmaceutických pomocných látek určitým technologickým postupem, uzpůsobený k tomu, aby léčivo, které obsahuje, mohlo vstoupit do interakce s organismem. Splňuje požadavky aplikace. [2]

1.1.3 Lékové formy

Léky mají lékovou (aplikační) formu. Je daná potřebou podání (užití, použití) léku a koexistencí v ní přítomných léčiv a pomocných látek. Do pojmu léková forma zahrnujeme nejen tvary léků, ale i jeho složení a fyzikální strukturu. [1]

Základní formy léků můžeme rozdělit do tří skupin

- tuhá forma (pevná)
- polotuhá
- tekutá forma.

1.1.3.1 Pevné lékové formy

- Tablety (tabulettae) vznikají slisováním prášků do různé velikosti. Tvary mohou být ploché válcovité, čtvercové, obdélníkové, čočkovité a další. Tablety jsou s dělicími rýhami a bez nich, jednovrstvé a dvouvrstvé, obalované a neobalované tablety. Např. Paralen je bílá tableta s půlicí rýhou a velikostí 18x8mm. [10]



Obrázek 1: Paralen

- Granuláty, zásypy, prášky pro injekce
Granule (granula) jsou léky ve formě prášku, stlačené do drobných, nepravidelných tvarů. Dávkování – po lžičkách. (např. Dologran). Velikost granulí je rozdílná, většinou okolo 3mm.



Obrázek 2: Dologran

- Šumivé tablety (tabulettae effervescens)

Léky jsou slisované do kotoučů, rozpustné ve sklenici vody (např. Celaskon). Celaskon je kulatá, růžová, mramorovaná, hladká, hygroskopická tableta pomerančové chuti o průměru cca 20mm.



Obrázek 3: Celaskon

- Kapsle, tobolky (capsules) - léky uvnitř želatinového obalu. Tento obal způsobuje zpomalení uvolňování léků v zažívacím traktu (např. Geriavit). Tato tobolka podlouhlého tvaru měří 25mm.



Obrázek 4: Geriavit

- Dražé (tabulettae obductae) jsou léky, které jsou obalené sladkým barevným povlakem, který chrání zubní sklovinu. Tyto léky jsou určeny k rozložení až v nižších částech trávicí trubice (např. Wobenzym).



Obrázek 5: Wobenzym

- Čípky (supositoria) - jedná se o léčebné látky, které jsou upravené do tvaru válce nebo kužele. Čípky dělíme podle způsobu aplikace na rektální a vaginální. Jsou vyráběné z kakaového másla, glycerinové želatiny. (např. Dobexil H)
Poševní globule (globuli vaginales) – jsou tvaru kulovitého, průměr 20 mm, určené k aplikaci do pochvy (např. Gyno – pevaryl).

[6][7]



Obrázek 6: Dobexil H

1.1.3.2 Polotuhé lékové formy

- Masti (unguenta) – lipoidní přípravky, které jsou určeny pro využití v kožním lékařství (např. Calcium pantotenicum)
- Pasty (pastae) – vznikají přidáním prášku do masťového základu. Pasty chrání kůži před mokváním.
- Gely (linimenta) – vzniká po smíchání léčivé látky s gelem. Nanáší se na kůži (např. Emoxen).
- Krémy (cremae) – hydrofilní přípravky k ochraně pokožky.

[6][7]

1.1.3.3 Kapalné lékové formy

- Roztoky, kapky, sirupy, suspenze pro vnitřní užití
- Kloktadla a spreje k aplikaci v ústech
- Roztoky, emulze, spreje, pěny na kůži
- Oční kapky a suspenze
- Rektální klyzmata a vaginální tekuté léky
- parenterální lékové formy - injekce a infuze

Roztoky (solutiones) – jsou vyrobeny z látek rozpustných v kapalině (fyziologický roztok), aplikují se po kapkách (guttae) do úst, spojivkového vaku, do nosních dutin, do ucha (např. Ophthalmo – Septonex, oční kapky, roztok)

Suspenze (suspensiones) – léčivá látka smíchaná s gelem, aplikuje se ústy k ochraně žaludeční sliznice (např. Anacid) [6] [7]



Obrázek 7: Tvary tablet



Obrázek 8: Ukázka použitých cukrovinek pro simulaci léčiv

1.2 Analýza použitých vývojových prostředků

1.2.1 Historie LEGO

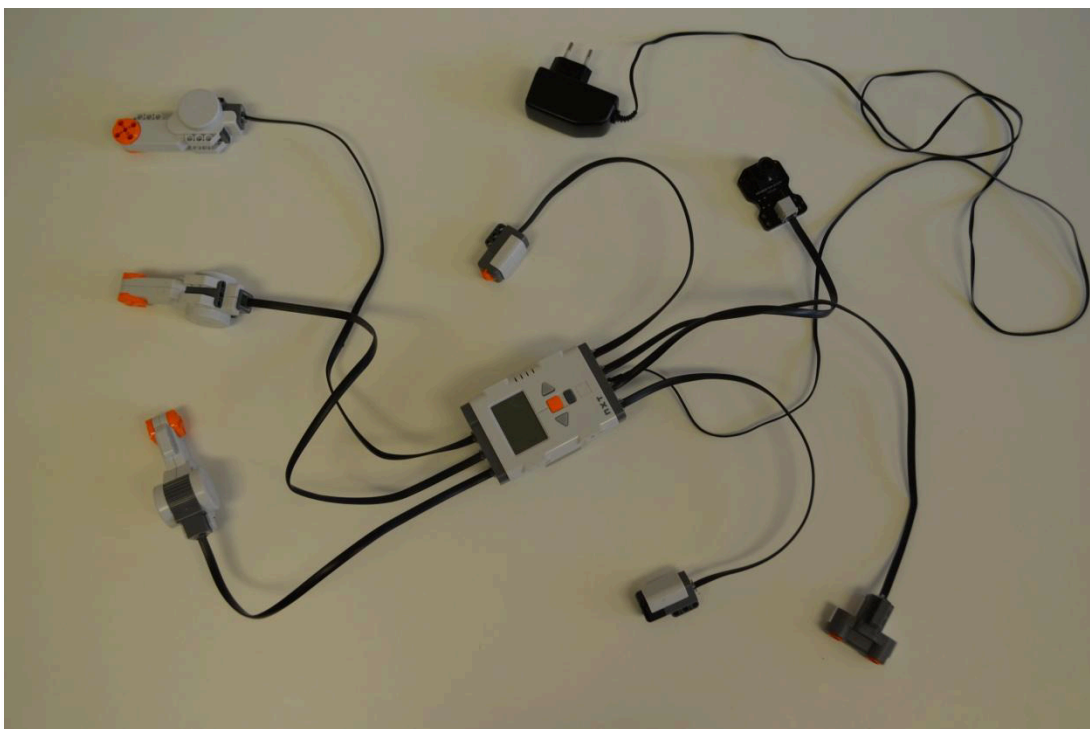
Název "LEGO" je zkratkou dvou dánských slov "leg godt", která znamenají "dobrou hru". Společnost LEGO byla založena Kirkem Kristiansenem v roce 1932. Tato společnost rozvíjí tvořivost dětí pomocí hry. Známost se stala díky svým stavebnicím, které se stále zdokonalují. Základem však stále zůstává LEGO kostka. LEGO kostka byla ve své stávající podobě uvedena na trh v roce 1958. Od roku 1983, kdy vzniká LEGO Duplo kostka, pokrývá LEGO stavebnicový systém prakticky celé rozpětí dětských let. Princip spojování kostek je jedinečný a nabízí nekonečné množství stavění. Je to jen otázka fantazie - a bohatství tvořivých nápadů ve hře. [9]



Obrázek 9: LEGO

V roce 1998 byla na trh uvedena stavebnice s mikropočítačem a byla nazvána MINDSTORMS. Obsahovala programovatelnou kostku RCX, senzory, interaktivní servomotor a propojovací kabely. V roce 2006 firma LEGO uvedla na trh novou řadu MINDSTORMS NXT. Byla založena na stejném principu, jen byla modernější. Původní 8 bitový mikroprocesor byl nahrazen 32 bitovým. V polovině roku 2009 byla verze LEGO MINDSTORMS NXT 2.0 aktualizována. K dispozici je nyní nová LEGO MINDSTORMS, tzv. EV3. EV3 oproti předchozí verzi má navíc wifi připojení, což usnadňuje přenos dat. Má o jeden výstupní port navíc a obsahuje port na kartu SD. [9]

1.2.2 Popis modulů NXT



Obrázek 10: Základní připojení vstupních a výstupních zařízení k portům řídicí jednotky NXT

1.2.2.1 Řídicí jednotka NXT (Řídicí kostka NXT)

Je hlavním aktivním prvkem stavebnice LEGO NXT. Řídicí kostkou není nazvaná jen díky svému tvaru, ale i kvůli specializovaným a univerzálním modulům, které mezi sebou komunikují. Do paměti procesoru řídicí jednotky NXT jsou nahrávána data, které jsou posléze zpracována. Pro nahrání dat do řídicí jednotky NXT slouží USB port typu B, pomocí kterého komunikuje s počítačem. Další možností komunikace je bezdrátové bluetooth rozhraní, přes které je možno zaslat data, ale i komunikace s více kostkami. [8]

Technická specifikace (verze NXT)

Hlavní procesor

- 32bitový mikroprocesor ARM7 (Atmel AT91SAM7S256)
- 48 MHz, 256 KB FLASH, 64 KB RAM

Přídavný koprocesor

- 8bitový mikroprocesor AVR (Atmel ATmega48)
- 8 MHz, 4KB FLASH, 512B RAM

Vstupy/ Výstupy porty

- 4 vstupní porty (6vodičový LEGO RJ12 konektor, IIC rychlost 9600 bit/s)
- výstupní porty s možností vstupu z kodéru (6vodičový LEGO RJ12 konektor)
- tlačítka na modulu pro ovládání uživatelského rozhraní a programů
- 8bitový zvukový výstup s reproduktorem (přehrávaná frekvence 2 – 16 kHz)
- Černobílý maticový displej 100 x 64 pixelů o velikosti 40,6 x 26 mm (programově použitelné 100 x 60 px)

Možností připojení

- USB 2.0 port (12 Mbit/sec)
- Bezdrátová Bluetooth® komunikace v2.0 třídy II, CSR BlueCore 4 v2.0 + EDR, 26MHz

Operační systém

- Operační systém vlastní (proprietární)

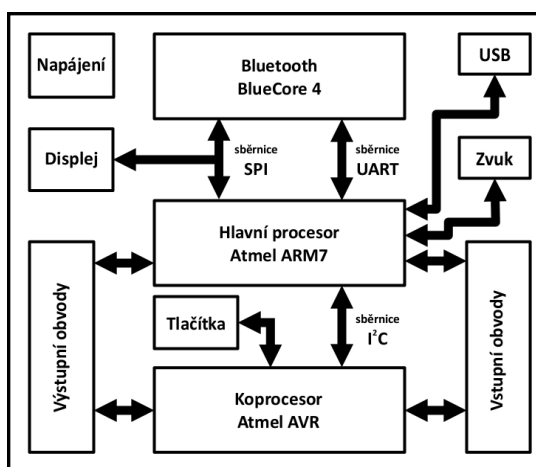
Zdroje energie

- 6 AA článků nebo nabíjecí lithiová baterie s transformátorem dle národní normy (EU: 230 V, 50 Hz)

[8]



Obrázek 11: NXT kostka



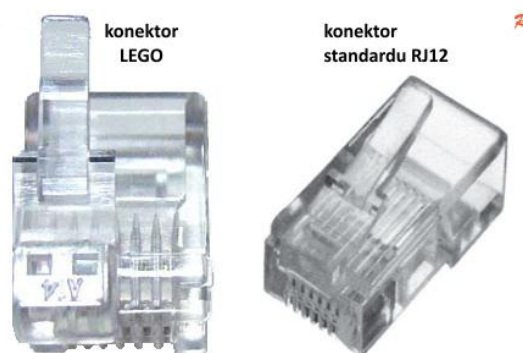
Obrázek 12: Vnitřní propojení modulu

1.2.2.2 Spojovací kabely NXT

Standardizované spojovací kabely NXT jsou naprosto nezbytné pro jakoukoliv práci se stavebnicí LEGO MINDSTORMS NXT. Používají se k připojení všech modulů, senzorů a servomotorů k řídicí jednotce NXT. V základní stavebnici se nachází sada sedmi těchto kabelů. Koncovka LEGO spojovacích kabelů má oproti standardnímu RJ12 telefonnímu kabelu posunutou horní část, která slouží k přidržení kabelu v zásuvce. Výrobce uvádí, že to je z důvodu bezpečnosti, aby nešli kabely zaměnit. Jinak se tyto dva kabely nijak neliší. Kabely se standardně vyrábějí ve třech délkách: 20, 35 a 50 cm. [8]



Obrázek 13: Spojovací kabel NXT



Obrázek 14: Rozdíl koncovek

1.2.2.3 Senzory

Robot dostává informace pomocí senzorů, které jsou k řídicí jednotce NXT připojeny čtyřmi vstupními porty. Senzory můžeme rozdělit do dvou skupin.

- Aktivní – světelný senzor
- Pasivní – zvukový senzor, dotykový senzor

[8]

1.2.2.3.1 Světelný senzor

Světelný senzor umožňuje rozlišovat světlo a tmu, ale může měřit i intenzitu odraženého světla. Intenzita barevných odstínů je udávána v procentech. Světelný senzor je umístěn v krytu a připojuje se pomocí spojovacích kabelů NXT k portu.

Světelný senzor může být použit jako měřič světelné úrovně světla a vyhodnocovat překročení intenzity světla, kterou předem můžeme nastavit. Další význam tohoto světelného senzoru je v rozpoznávání barevného podkladu. [8]



Obrázek 15: Světelný senzor

1.2.2.3.2 Dotykový senzor

Dotykový senzor funguje jako tlačítko vracující logickou hodnotu pravda/nepravda (true/false). Dotykovému senzoru jsou přiřazeny tři akce.

- Zmáčknutí (Pressed)
- Uvolnění (Released)
- Zmáčknutí a uvolnění (Bumped)

Dotykový senzor může sloužit jako tlačítko pro spouštění či ovládání programu (zatačení vlevo nebo vpravo) nebo jako senzor doteku (náráz do překážky, zjištění přítomnosti objektu).

[8]



Obrázek 16: Dotykový senzor

1.2.2.3.3 Barevný světelný senzor

Umožňuje rozpoznat až 15 barev. Barevný světelný senzor je namířen na povrch a pomocí bílé LED diody, která osvětlí povrch předmětu, analyzuje barevné spektrum světla, které se odrazilo od povrchu. Poté vrací číslo barvy, která je podle továrního nastavení nejbližší. Při měření je barevný světelný senzor umístěn několik centimetrů od předmětu a pod úhlem, aby se zabránilo odrazu. Senzor vrací čísla barev od 0 – 17. Černá barva je pod číslem 0 a bílá pod 17. (viz Obrázek 33). Modul je standardně připojen pomocí spojovacího kabelu NXT a jeho obal je stejný jako u všech ostatních rozšiřujících modulů. Senzor pak snímá a vrací číslo nové barvy přibližně 100x za sekundu. [8]



Obrázek 17: Barevný světelný senzor

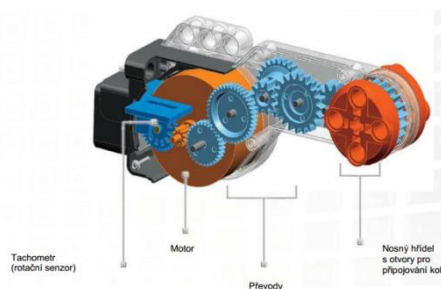
1.2.2.4 Interaktivní servomotory

Interaktivní servomotor je využívané výstupní zařízení stavebnice LEGO MINDSTORMS. Pomocí servomotorů se uvádějí do pohybu pohyblivé části. (auta, vozíky, stroje, posuvné pásy). Dá se využít i jako vstupní zařízení, které informuje o úrovni natočení rotačního senzoru uvnitř servomotoru. Umožňuje přesnější ovládání robota.

K řídicí jednotce NXT lze připojit do výstupních portů až tři servomotory, které se dají řídit nezávisle nebo spřaženě. Lze použít jako pohonná jednotka, k přemístění, zvednutí nebo k přesunutí z místa na místo. [8]



Obrázek 18: Interaktivní servomotor



Obrázek 19: Vnitřní uspořádání interaktivního servomotoru

1.2.2.5 NXT kamera

NXT kamera slouží k reálnému zpracování obrazu. Kamera je přístupná přes standardní NXT senzor port, ale je ji možné připojit USB kabelem s PC. Zpracovaná informace obsahuje ohraničený rámeček polohy objektů. NXT kamera – v4 je kompatibilní s NXT kamerou – v3 softwarem i knihovnamí.



Obrázek 20: NXT kamera

Specifikace NXT kamery:

- Sledování až 8 různých barevných objektů při rychlosti 30snímků / sekundu.
- Konfigurace NXT kamery pomocí rozhraní USB v systému Windows XP, Vista nebo 7.
- Podporuje dva sledovací režimy: sledování objektů a čar.
- O sledovaných objektech zjistí statistiky (např. počet objektů, barvu předmětu, ohraničující okno polohy nebo souřadnice řádku) prostřednictvím standardního NXT senzor portu.
- Pásové rozlišení obrazu je 88x144 pixelů při rychlosti 30snímků / sekundu.
- Barevný obraz zobrazí v plném rozlišení (176x144pixelů) na PC přes USB port.
- Maximální příkon 42mA při 4,7V .

NXT používá kompatibilní I2C protokol pro komunikaci.

[12]

1.3 Analýza zpracování digitálního obrazu

Obraz může být pořízen digitálním fotoaparátem v různých oblastech viditelného spektra, jako je třeba denní světlo. Může být ale pořízen i v oblastech záření, které lidské oko nevidí (infračervené snímky, rentgenové snímky). Světlo je elektromagnetické vlnění, které není ve skutečnosti ničím jiným než dvojdimenzionálním signálem. Obrazy snímáné v různém spektru zachycují obsah snímku zcela jinak. Vlnová délka záření, při které je snímek snímán, musí být navíc menší než zkoumaný detail. [3]

1.3.1 Digitalizace

Čidla pro vstup obrazové funkce jsou většinou zdrojem spojitého signálu. Abychom obrazovou funkci mohli zpracovat v počítači, musíme ji digitalizovat. Digitalizace spočívá ve vzorkování obrazu v matici $M \times N$ bodů a v kvantování spojitě jasové úrovně každého vzorku do K intervalů. Mezi nejběžnější způsoby digitalizace obrazu patří snímání digitálním fotoaparátem a skenování. Jako snímače obrazu jsou dnes nejvíce rozšířeny obrazové senzory nazývané dle technologie a způsobu transportu informace jako CCD nebo CMOS. [3, 4]

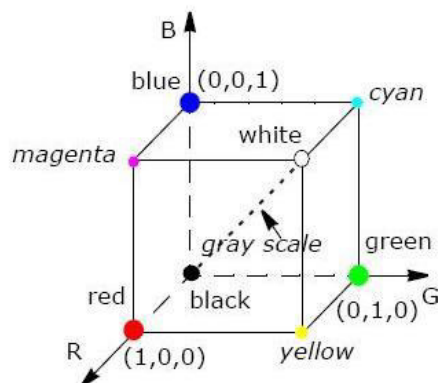
1.3.2 Barevné modely

Barva a barevné modely hrají ve zpracování obrazu a fotografie velkou roli. Barevné světlo vnímané lidským zrakem má rozsah spektra přibližně od 400 do 700nm. Pokusy s rozkladem slunečního světla pomocí skleněného hranolu prováděl již Isaac Newton v roce 1666. [3]

1.3.2.1 Barevný model RGB

Model RGB (Red, Green, Blue – červená, zelená, modrá) se nejčastěji používá pro snímání barevného obrazu a zobrazování na monitoru. Barevný model RGB byl standardizován již v roce 1931 komisí CIE. Tento obraz dle tohoto standardizovaného modelu je složen aditivně ze tří barev: červené (red), zelené (green), modré (blue). Jednotlivým složkám byly přiřazeny vlnové délky: red = 700nm, green = 546,1nm, blue = 435,8nm. Z technického hlediska je každá barva reprezentována určitým počtem bitů

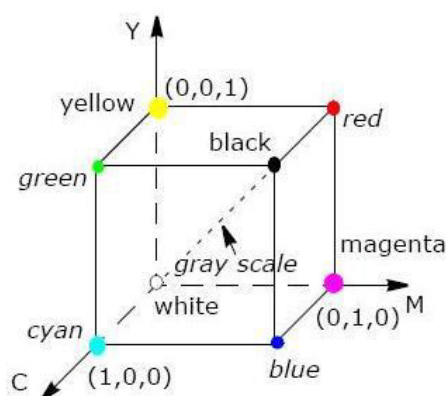
(běžně to bývá 8bitů). Barevný prostor RGB se označuje jako aditivní skládání barev (čím vyšší hodnoty, tím světlejší). [3]



Obrázek 21: Model RGB

1.3.2.2 Barevný model CMY a CMYK

Barevný model CMY (Cyan, Magenta, Yellow – modrozelená, purpurová, žlutá) se používá pro tisk. Říká se mu subtraktivní. U subtraktivního modelu odpovídá počátek souřadnic bílé a vrchol [1, 1, 1] černé. Barvy modrozelená C, purpurová M a žlutá Y leží na krychli v protilehlých rozích oproti RGB. Překrýváním těchto tří barev nevznikne dokonalá černá, proto byl zaveden model CMYK, kde K znamená černou (black). [3]



Obrázek 22: Model CMY

2. NÁVRH METOD PRO ANALÝZU TVARU, VELIKOSTI A BAREV Z OBRAZOVÉHO SIGNÁLU

2.1 Programování třídící linky

Ovládání řídicí jednotky NXT LEGO MINDSTORMS lze provádět buďto přímo v menu řídicí jednotky NXT či pomocí programu vytvořeného v programovacím prostředí, který má obvykle značně více možností. Specifickým způsobem je ovládání řídicí jednotky NXT přes mobilní telefon a jeho bluetooth rozhraní, které se hodí buďto pro názorné vysvětlení základních principů práce s motory a znázornění výsledné měřené hodnoty senzory nebo pro úlohy, jenž z důvodu lepší použitelnosti nemohou obsahovat kabely. [8]

programovací prostředí	NXC	RobotC	Matlab
Jazyk	C	C	Matlab toolbox
Platformy	Windows, Linux	Windows	Windows
obsluha událostí	Ne	Ano	-
Multivláknové aplikace	Ano	Ano	-
podpora bluetooth	Ano	Ano	Ano

Tabulka 1: Přehled možností programování robota

[13]

2.1.1 Matlab

Matlab je programovací jazyk na vysoké úrovni pro numerické výpočty, získávání a analýzy dat. Může být použit pro ovládání LEGO MINDSTORMS NXT přes Bluetooth sériový port (sériový port komunikace je součástí základní funkčnosti Matlab) nebo přes USB připojení: například pomocí MINDSTORMS NXT Toolbox.

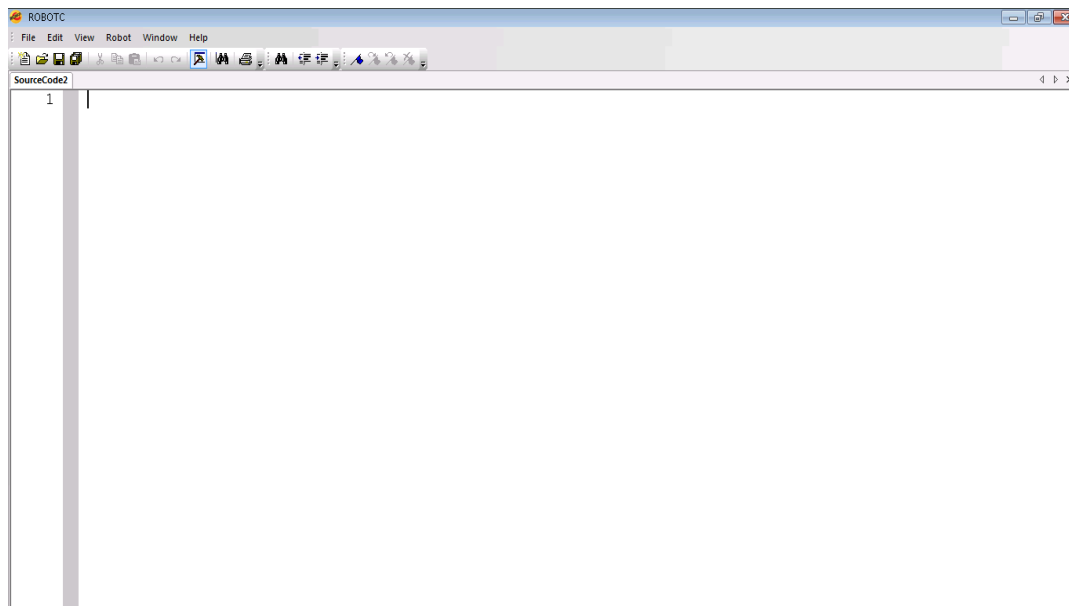
2.1.2 NXC

Textový jazyk odvozený od jazyka C běží v prostředí BricxCC na standardním firmware LEGO MINDSTORMS.

2.1.3 RobotC

RobotC je jazyk pro programování stavebnic LEGO MINDSTORMS NXT v jazyce C. Programování je prováděno textovým zápisem příkazů jazyka C v kombinaci s příkazy získávající informace ze senzorů a ovládající výstupní zařízení. RobotC má stejné názvy příkazu jako jiné profesionální nástroje používané v jiných programovacích jazycích. K RobotC existuje řada vzorových programů s rozsáhlou dokumentací, což usnadňuje programování.

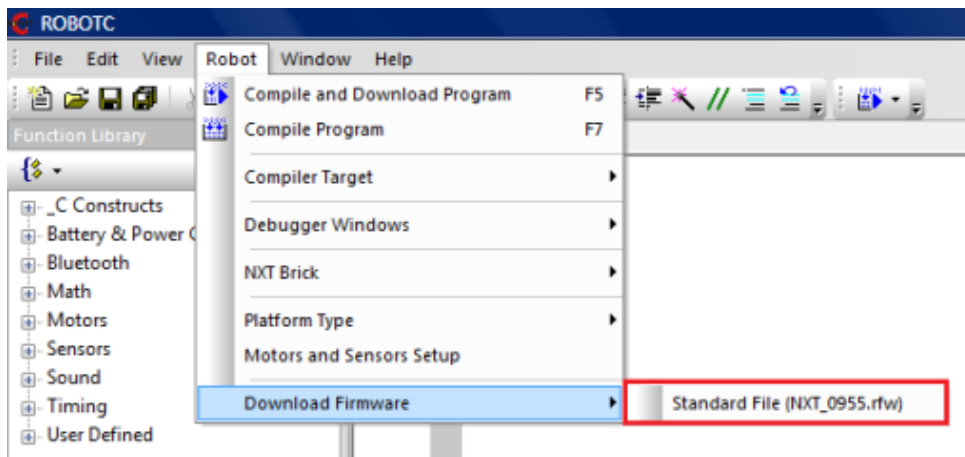
S ohledem na zadání mé bakalářské práce mi více vyhovuje pro psaní algoritmu RobotC. [8, 11]



Obrázek 23: Ukázka rozhraní v programu RobotC

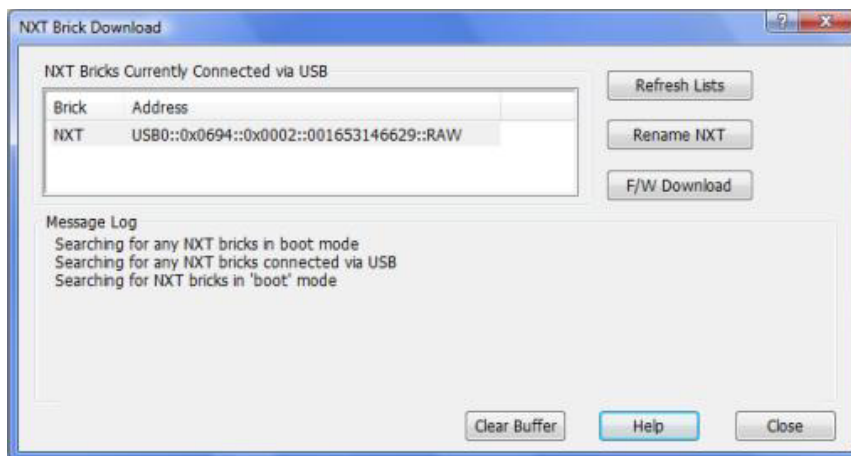
2.1.3.1 Postup nahrání firmware

Pro přečtení algoritmu pomocí řídicí jednotky NXT bylo potřeba vyšší verze firmware, proto se musel vždy aktualizovat. Pro aktualizaci byl použit USB kabel typu B přiložený u stavebnice LEGO MINDSTORMS. Postup pro aktualizaci byl proveden v nabídce hlavního menu programu Robot → Download Firmware → Standard File



Obrázek 24: Aktualizace firmware

Dále se zobrazí nové podokno pro aktualizaci, kde bylo zvoleno F/W Download.

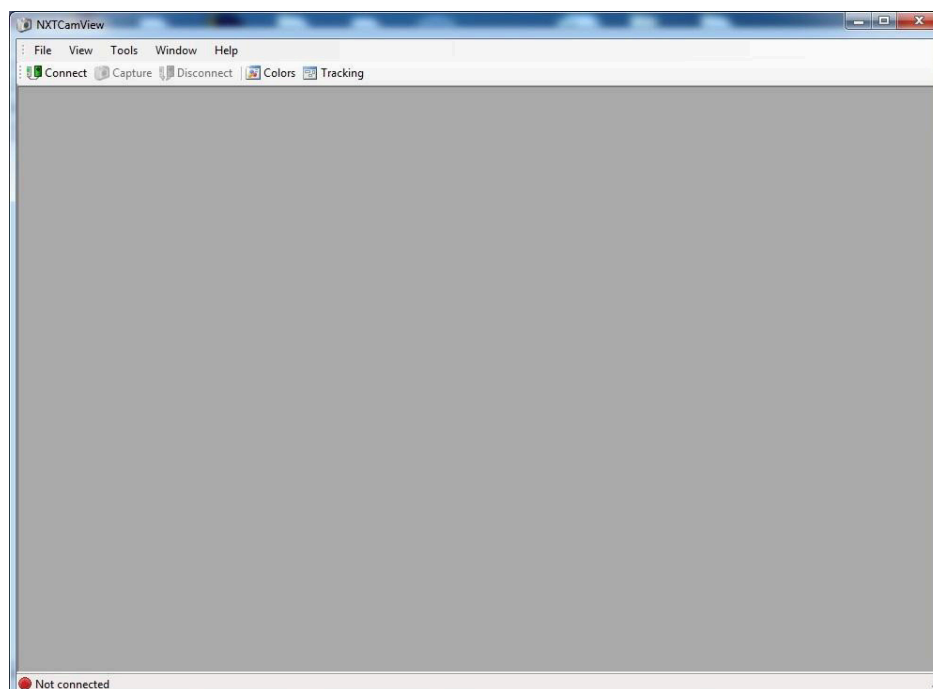


Obrázek 25: Nahrání firmware

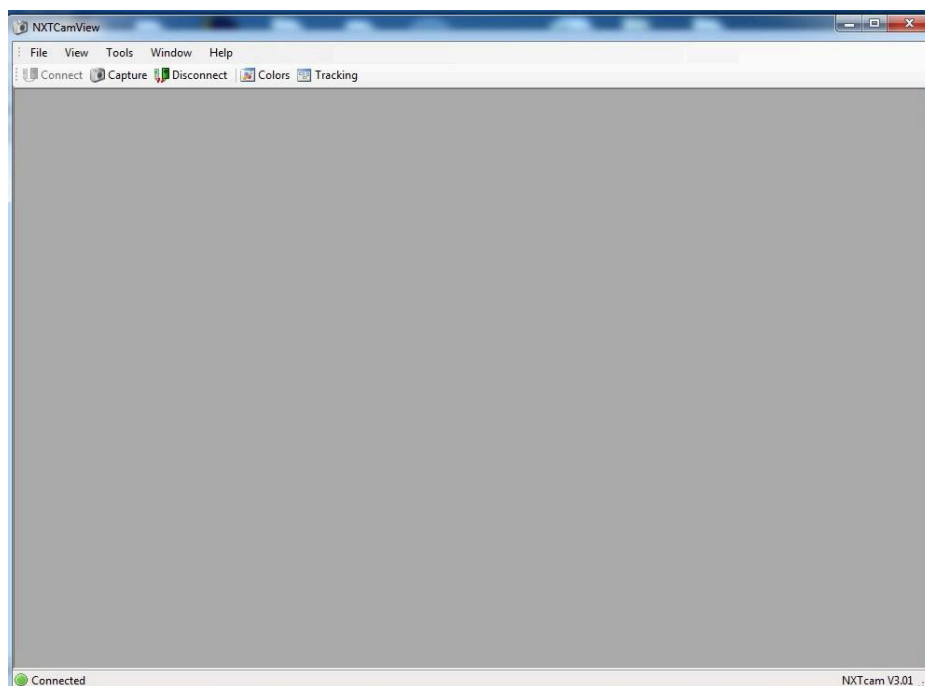
2.1.4 NXTCamView

NXT CamView konfiguruje a ovládá NXT kameru. NXTCamView bylo nutné nejdříve stáhnout a nainstalovat na PC. Dostupný ke stažení je na webových stránkách: <http://nxtcamview.sourceforge.net/>. Po nainstalování a spuštění programu NXTCamView, se zobrazí úvodní stránka, která je zobrazena na obrázku 26. Dále bylo nutné spojit NXT kameru pomocí USB portu k PC. Následně bylo potřeba stisknout ikonu Connect pro připojení NXT kamery k PC (viz obrázek 27). Stiskem ikony Capture se nasnímal obraz tablety se kterým lze dále pracovat (výběr barevné škály).

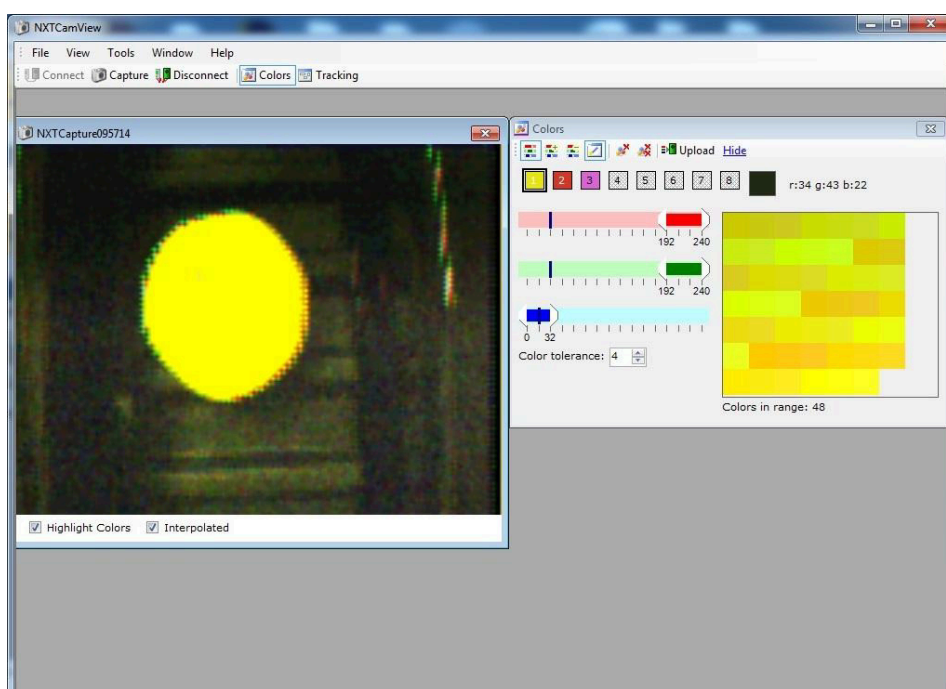
Hlavním cílem NXTCamView je naučit NXT kameru, které barvy má rozpoznat a sledovat, pro detekci objektu v jeho pohledu. Barvy jsou zobrazeny jako rozmezí červené, zelené a modré (tedy jejich minimální a maximální hodnoty každé z nich). Vybranou barevnou škálu je nutno nahrát do NXT kamery pomocí ikony Upload.



Obrázek 26: úvodní stránka NXTCamView



Obrázek 27: NXTCamView připravený k načtení objektu



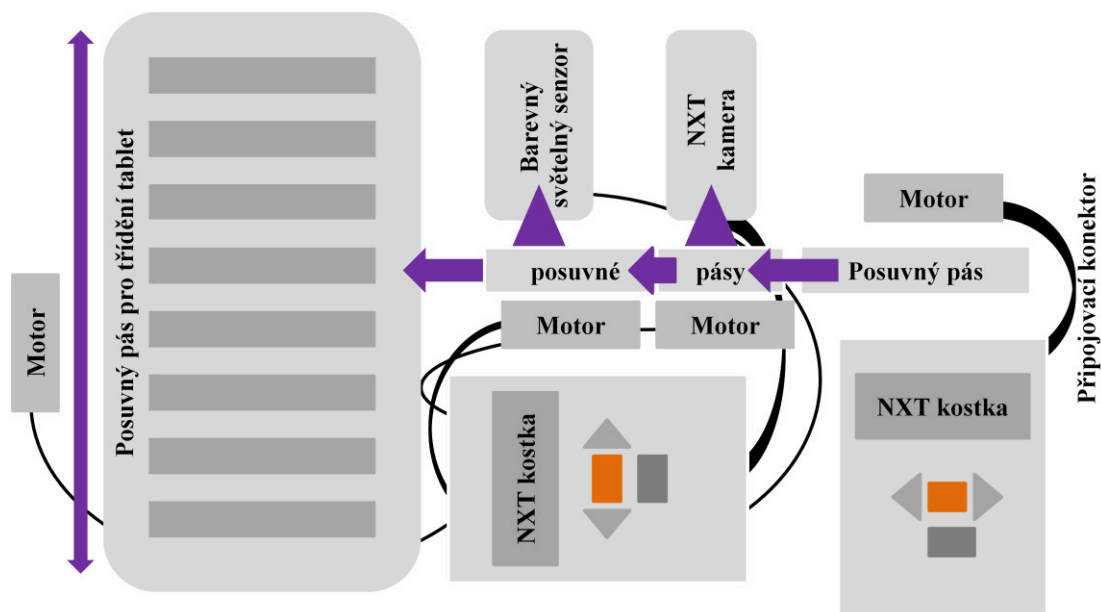
Obrázek 28: načtený objekt v programu NXTCamView

3. NÁVRH A REALIZACE ŘEŠENÍ

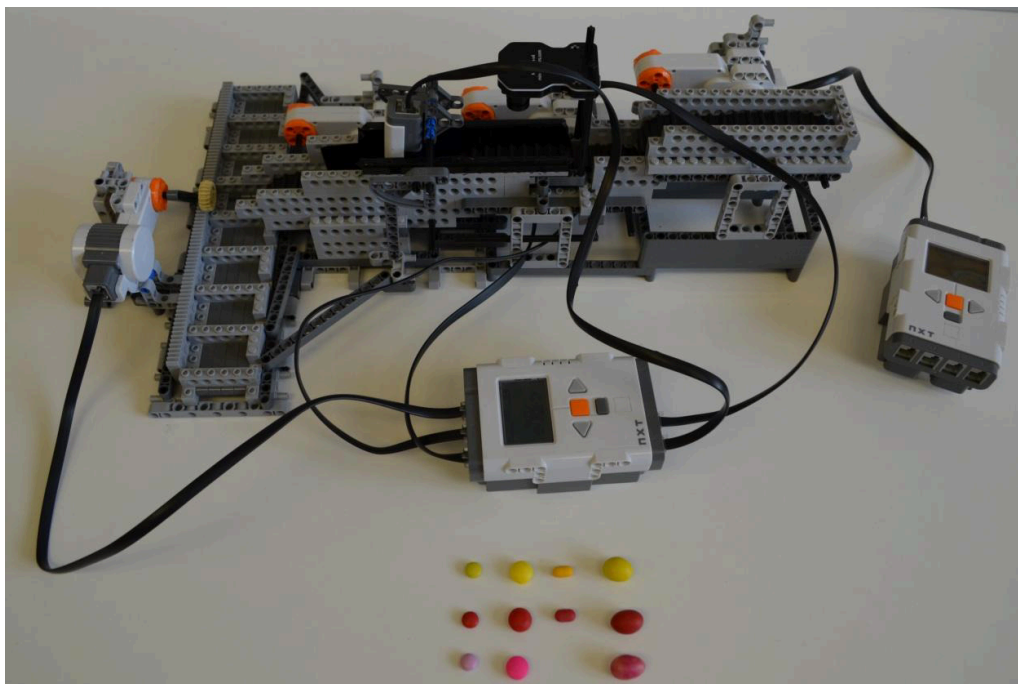
Návrh realizace třídící linky je složen s dvou řídících jednotek NXT a dalších modulů. To jsou 4 interaktivní servomotory, NXT kamera, barevný světelný senzor, spojovací kabely NXT a stavebnice LEGO.

Toto schéma na obrázku 29 ukazuje, jak jsem třídící linku na automatizovanou detekci a třídění typu tablet navrhla. Bude obsahovat posuvný pás na tablety, který bude postupně posouvat různé typy tablet na následující posuvný pás. Posuvné pásy budou poháněny pomocí interaktivních servomotorů. Tablety budou nejprve směřovat pod NXT kameru. NXT kamera rozezná velikost a tvar tablet. Poté budou směřovány pod barevný světelný senzor, který určí jejich barvu. V konečné fázi budou následně tablety zařazeny do zásobníku pro třídění tablet. Zásobník je navrhnut jako posuvný, rozmístí tablety podle jejich tvaru, velikosti a barvy. V mém řešení bakalářské práce je místo tablet využito různých forem cukrovinek (lentilky, Tic Tac, skittles, m&m's).

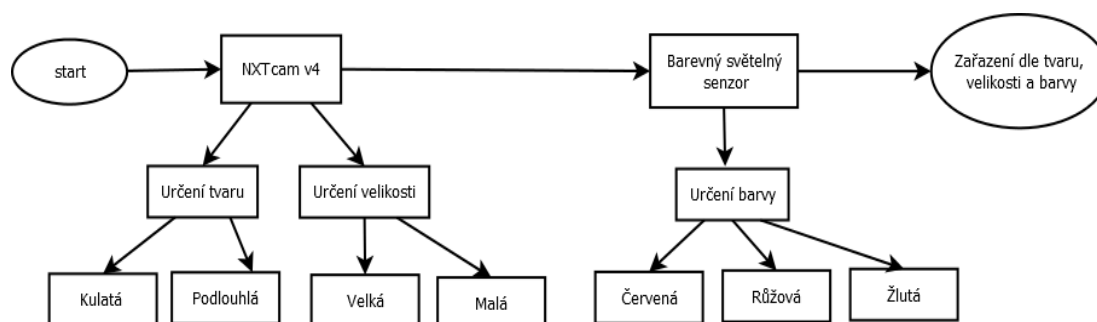
V této kapitole se budu zabývat sestavením a programováním jednotlivých komponentů přístroje LEGO MINDSTORMS. K programování používám programovací rozhraní programu RobotC for MINDSTORMS 2.02. Přenos dat probíhá pomocí USB rozhraní 2.0.



Obrázek 29: Návrh řešení třídící linky



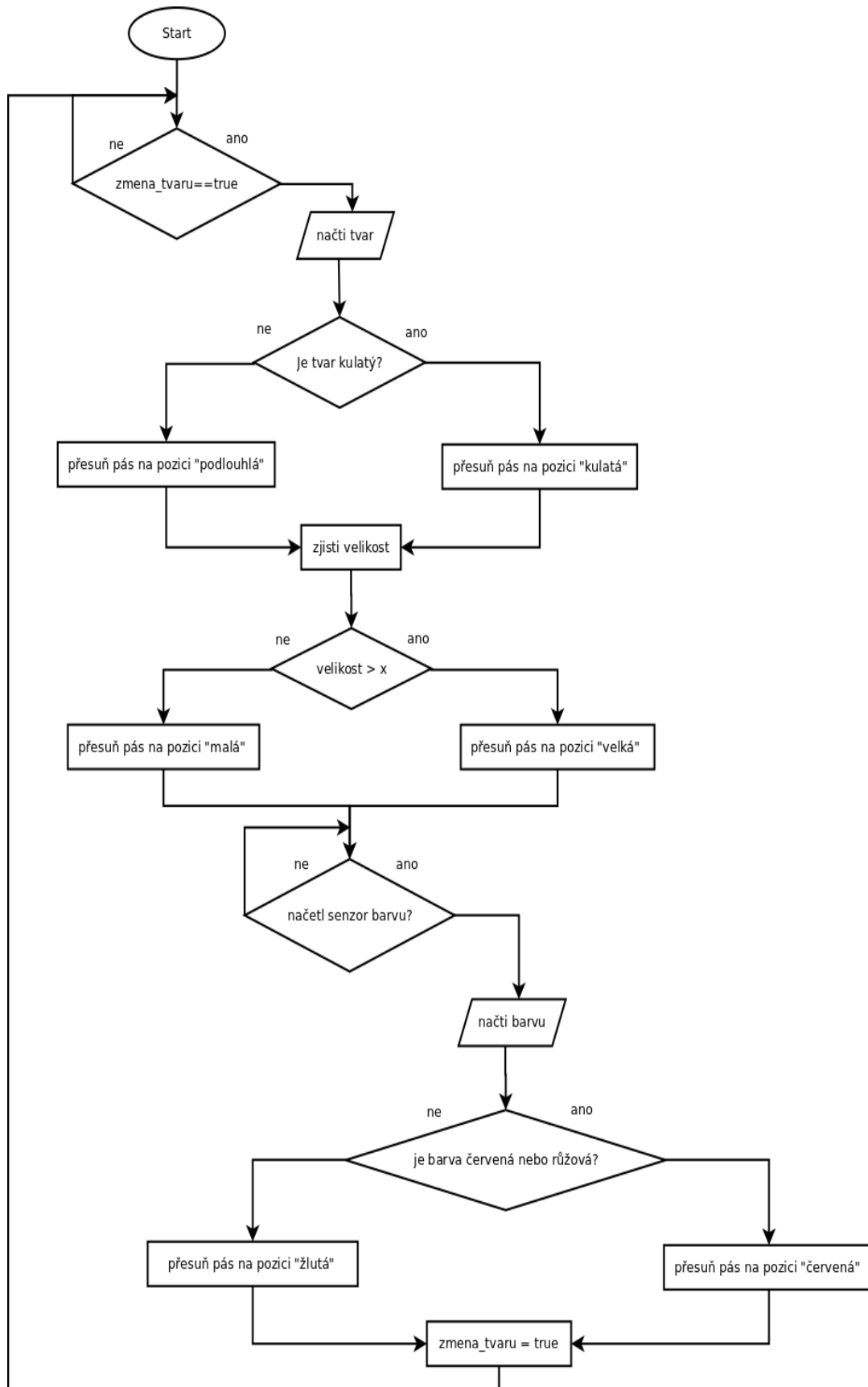
Obrázek 30: Realizace řešení třídící linky



Obrázek 31: Princip algoritmu třídící linky

Vývojový diagram třídící linky

Na obrázku 32 je vývojový diagram třídící linky, který popisuje automatizovanou detekci a třídění typu tablet. V prvním kroku musí NXT kamera zaznamenat přítomnost tablety a poté nasnímá její tvar. Pokud je tvar kulatý posune se zásobník na pozici „kulatá“, pokud ne posune se na pozici „podlouhlá“. Dále se zjistí velikost, pokud je velikost větší jak hodnota x posune se na pozici „velká“, pokud ne posune se na pozici „malá“. V dalším kroku musí barevný světelný senzor zaznamenat přítomnost změny barvy a načíst jí. Pokud je červená nebo růžová posune se na pozici „červená“, pokud ne posune se na pozici „žlutá“. Poté se celý postup opakuje.



Obrázek 32: Vývojový diagram třídící linky

3.1 Vytvoření třídícího mechanismu s použitím interaktivních servomotorů

Interaktivní servomotor slouží pro chod posuvných pásů pro přepravu tablet. V třídící lince mám tři posuvné pásy ovládané servomotory pro přepravu tablet mezi jednotlivými senzory a posuvný zásobník pro detekované tablety.

V programu RobotC jsem vytvořila nový soubor, do kterého jsem vepsala kód pro rozběhnutí servomotorů vpřed. Tento kód slouží pro přepravu tablet pro posuvné pásy A a B. Pás C slouží pro zařazení tablet podle jejich tvaru, velikosti a barvy. Dále jsem použila novou řídicí jednotku NXT, jelikož každá řídicí jednotka NXT disponuje třemi porty pro výstup servomotorů. Tato řídicí jednotka slouží pouze pro pás A, který spouští tabletu na třídící pásy po určitém časovém intervalu.

3.1.1 Kód použitý v programu RobotC pro motor posuvného pásu A

```
task main() {
    for(;;) {
        motor[motorA] = -5;
        wait1Msec(1300);
        motor[motorA] = 0;
        wait1Msec(20000);
    }
}
```

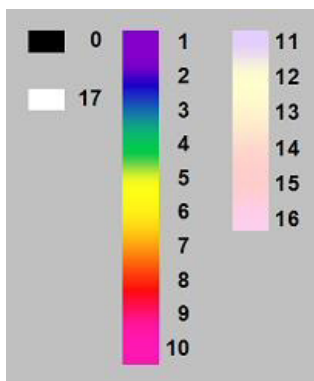
3.1.2 Kód použitý v programu RobotC pro motor posuvného pásu A a B

```
void motor_pas_on() {
    motor[motorA] = -5;
    motor[motorB] = -5;
}
```

3.2 Vytvoření rozlišovacího mechanismu s využitím barevného světelného senzoru

Barevný světelný senzor je v třídící lince použit pro rozlišení barvy tablet.

Barevný světelný senzor může rozlišovat až 15 barev, ale je nutné ho kalibrovat. Senzor je kalibrován tak, že rozlišuje barvy podle barevné škály, uvedené na obrázku 33 v hodnotě od 0 do 17. 0 je přiřazená černá, 1 – 10 je základní barevná škála. 11 – 16 jsou odstíny šedi a 17 je bílá. Barva je na řídicí jednotce NXT displeje zobrazena jako číselná hodnota. Barva má přiřazenou číselnou hodnotu, kterou by bylo nutné vyhledat, proto jsem v dalším kroku rozšířila zdrojový kód, aby se na řídicí jednotce NXT displeje zobrazil i název barvy. V práci využívám tři barvy a to červenou, žlutou a růžovou. Růžová tableta je naprogramována tak, aby byla zařazena k červené tabletě v zásobníku.



Obrázek 33: Kalibrované barvy



Obrázek 34: Barevný světelný senzor

3.2.1 Načtení knihovny pro barevný světelný senzor

```
#include "HTCS-driver.h"
```

Knihovna pro barevný světelný senzor obsahuje funkce pro načtení barev v RGB škále a konstanty jednotlivých barev.

3.2.2 Kód použitý v programu RobotC pro načtení barvy

```
if(zmena_barvy) {
    color = HTCSreadColor(HTCS);
    nxtDisplayTextLine(0, "%d", color);
if (color!=12)
    {

        if(color==CERVENA || color==RUZOVA) {
            motor_cervena();
            zmena_barvy = false;
            last_color = color;

            if(color==ZLUTA1 || color==ZLUTA2) {
                motor_zluta();
                zmena_barvy = false;
                last_color = color;
            }
        }
    }
}
```

3.2.3 Kód použitý v programu RobotC pro zařazení barvy v posuvném zásobníku

```
void motor_zluta() {
    if(last_color==CERVENA || last_color==RUZOVA) {
        motor[motorC] = 17;
        wait1Msec(1100);
        motor[motorC] = 0;
    }
}

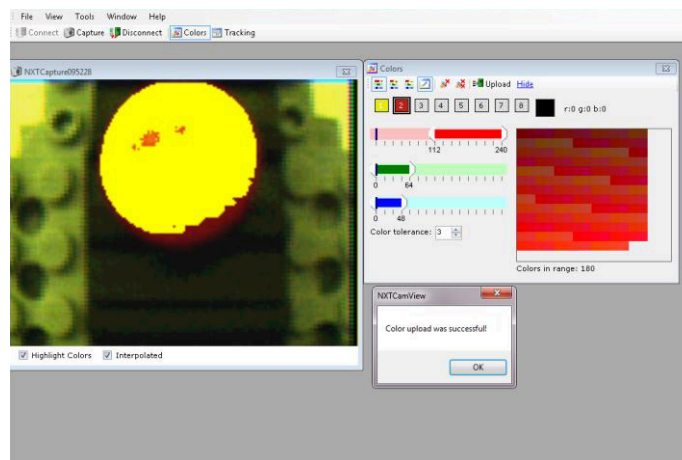
void motor_cervena() {
    if(last_color==0 || last_color==ZLUTA1 || last_color==ZLUTA2)
    {

        motor[motorC] = -17;
        wait1Msec(1100);
        motor[motorC] = 0;
    }
}
```

3.3 Vytvoření rozlišovacího mechanismu s využitím NXT kamery

NXT kameru využívám v mnou navržené třídící lince pro určení tvaru a velikosti tablet. Velikosti mám malé a velké. Tvar je kulatý a podlouhlý.

NXT kamera může rozlišovat až 8 barevných objektů. NXT kameru lze zaostřovat jemným pootočením objektivu. Pro ideální načtení objektu je vhodné používat bílou zářivku. V mých pokusech jsem pracovala pouze s přirozenými světelnými podmínkami, proto není vždy obraz ideální.



Obrázek 35: Ukázka načtení objektu v programu Camview

3.3.1 Načtení knihovny pro NXT kameru

```
#include "nxtcamlib.c"
```

Knihovna inicializuje kameru a umožňuje získání viditelných objektů na určité pozici.

3.3.2 Kód pro určení tvaru v programu RobotC

```

if((delta_x>delta_y&&(delta_x<delta_y+delta_x/3.5)) |
| (delta_y>delta
    nxtDisplayTextLine(0,"kulata");
    zmena_barvy = true;
    motor_ctverec(obsah);
}
else {
    nxtDisplayTextLine(0,"podlouhla");
    zmena_barvy = true;
    motor_obdelnik(obsah);
}
zmena_tvaru = false;
}
}

```

3.3.3 Kód pro určení velikosti a zařazení pro kulatou tabletu v programu RobotC

```

void motor_ctverec(float v) {
    if(tvar!=0) {
        motor[motorC] = 17;
        wait1Msec(4300);
        motor[motorC] = 0;
    }
    tvar = 0

    if(v<=1000&&velikost==1) {
        motor_velikost_mala();
    }
    if(v>1000&&velikost==0) {
        motor_velikost_velka();
    }
}

```

4. VERIFIKACE NAMĚŘENÝCH A ANALYZOVANÝCH VÝSLEDKŮ

Analýzu sestavené třídící linky ze stavebnice LEGO MINDSTORMS a sepsaného zdrojového kódu v programu RobotC for MINDSTORMS 2.02 jsem prováděla pomocí tablet dvou velikostí (malé a velké), dále dle tvaru (kulatá a podlouhlá) a barvy (červená, žlutá a růžová). Celkem jsem provedla 20 pokusů o celkovém počtu tablet 16. Do celkové tabulky jsem sepsala počet správně a špatně zařazených tablet. Dále jsem v tabulce uvedla počet chybných zařazení NXT kamery i barevného světelného senzoru, chybných zařazení pouze NXT kamery a chybných zařazení pouze barevného světelného senzoru. Následně jsem vypočítala úspěšnost správně a špatně zařazených tablet a zprůměrovala je.

číslo pokusu	celkem tablet	zařazené správně	zařazené špatně	chyba NXTkamery i barevného sv.senzoru	chyba NXT kamery	chyba barevného sv.senzoru	úspěšnost správně [%]	úspěšnost špatně [%]
1	16	11	5	1	2	2	68,75	31,25
2	16	10	6	1	5	0	62,50	37,50
3	16	9	7	2	4	1	56,25	43,75
4	16	12	4	1	3	0	75,00	25,00
5	16	7	9	1	3	5	43,75	56,25
6	16	10	6	0	3	3	62,50	37,50
7	16	12	4	0	0	4	75,00	25,00
8	16	10	6	0	3	3	62,50	37,50
9	16	13	3	1	1	1	81,25	18,75
10	16	13	3	0	0	3	81,25	18,75
11	16	12	4	0	3	1	75,00	25,00
12	16	15	1	0	0	1	93,75	6,25
13	16	14	2	0	0	2	87,50	12,50
14	16	16	0	0	0	0	100,00	0,00
15	16	12	4	0	1	3	75,00	25,00
16	16	13	3	1	1	1	81,25	18,75
17	16	12	4	0	2	2	75,00	25,00
18	16	16	0	0	0	0	100,00	0,00
19	16	14	2	0	0	2	87,50	12,50
20	16	14	2	0	0	2	87,50	12,50
průměr	16,00	12,25	3,75	0,40	1,55	1,80	76,56	23,44
průměr v %	100,00	76,56	23,44	2,50	9,69	11,25	-	-
počet tablet	320	245	75	8	31	36	-	-

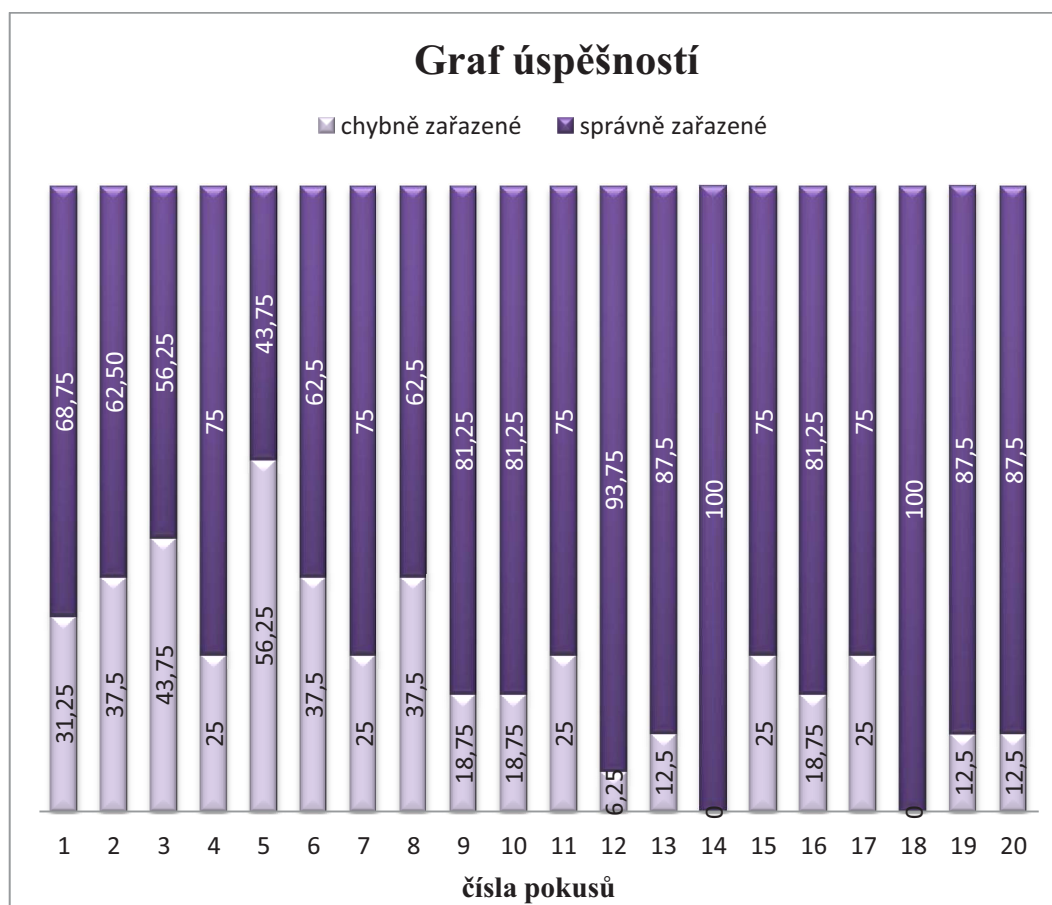
Tabulka 2: Naměřené hodnoty

4.1 Úspěšnost zařazení tablet

V tabulce 3 jsou procenticky vyjádřeny hodnoty správně a špatně zařazených tablet. Celková průměrná hodnota správně zařazených tablet je 76,56% a zbylých 23,44% reprezentují špatně zařazené tablety. Tabulka je doplněná grafem 1 úspěšnosti pro správně a špatně zařazené tablety. Dvě série pokusů proběhly se stoprocentně správnou úspěšností, tzn., že všechny tablety byly správně zařazeny do zásobníku.

číslo pokusů	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
chybně zařazené	31	38	44	25	56	38	25	38	19	19	25	6,3	13	0	25	19	25	0	13	13
správně zařazené	69	62	56	75	44	63	75	63	81	81	75	94	87	100	75	81	75	100	87	87

Tabulka 3: Chybně a správně zařazené hodnoty v %



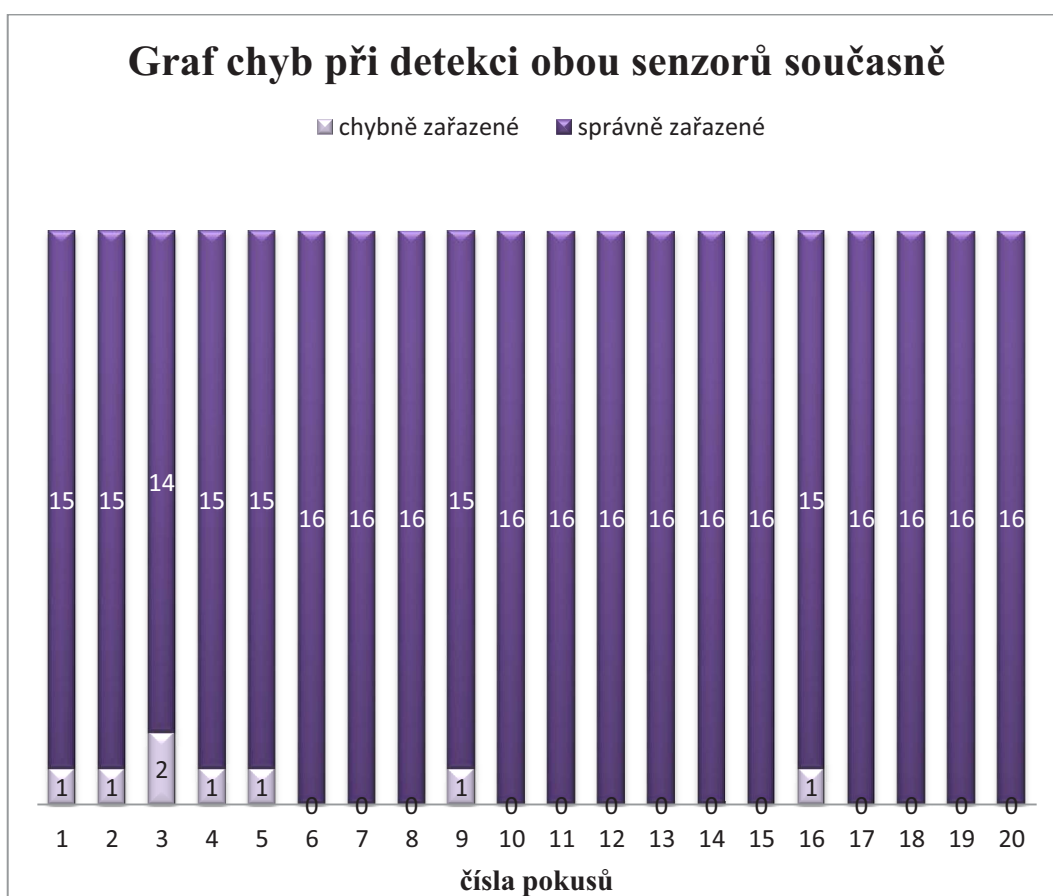
Graf 1: Graf úspěšnosti pro správně a špatně zařazené tablety

4.2 Četnost chyb při detekci obou senzorů

Tabulka 4 uvádí počet chyb při detekci obou senzorů (NXT kamera i barevný světelný senzor). Z celkového počtu 20 pokusů se v 7 pokusech vyskytla chyba současně u NXT kamery i barevného světelného senzoru. Z toho v 6 pokusech byla špatně zařazená oběma senzory současně 1 tableta z celkového počtu 16. V jednom případě 2. Souhrn výsledků je uveden v grafu 2. Z výsledků uvedených v celkové tabulce (tabulka 2) vyplývá, že oba senzory současně chybovaly průměrně ve 2,5 procentech. Chyby jsou odůvodněny níže v kapitole 4.6 a 4.7.

číslo pokusů	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
počet chyb obou senzorů současně	1	1	2	1	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0

Tabulka 4: Četnost chyb při detekci obou senzorů



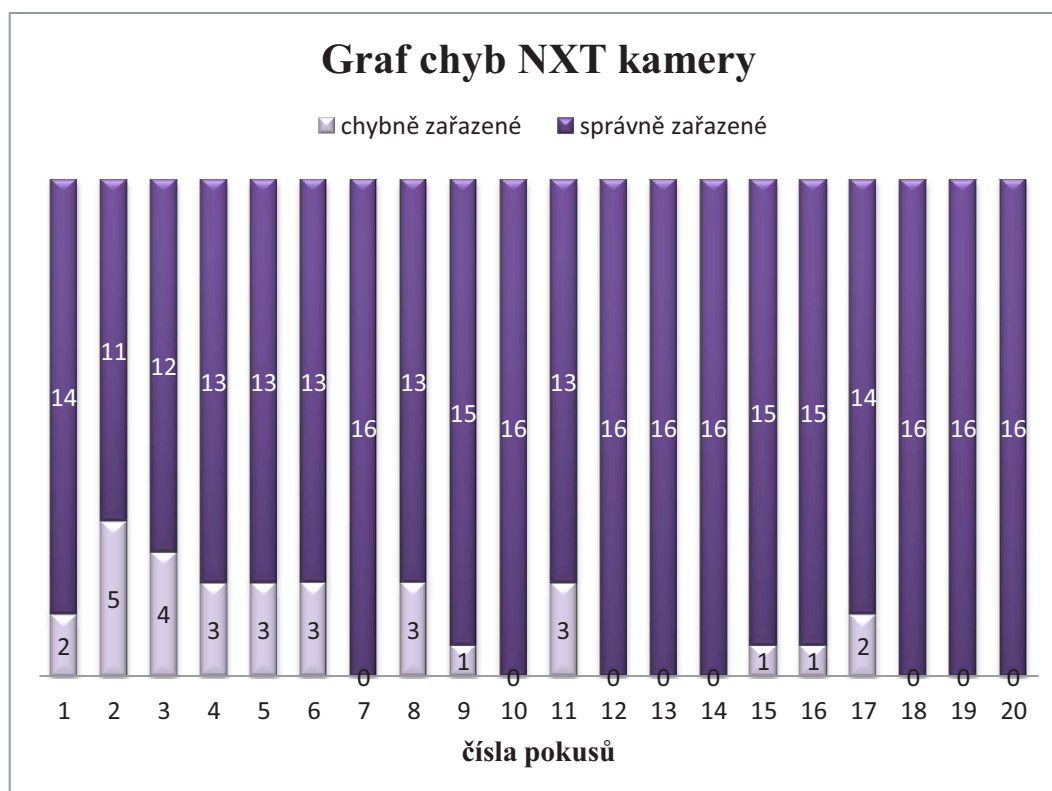
Graf 2: Současné zaznamenání chyb NXT kamery i barevného světelného senzoru

4.3 Četnost chyb při detekci NXT kamery

NXT kamera je v mnou sestavené třídící lince zařazena první v pořadí. Detekují se ji tvar a velikost tablet. Z provedených 20 pokusů (1pokus=16tablet) bylo pouze 8 pokusů bez chybného zařazení. V tabulce 5 je zobrazena četnost chyb při detekci NXT kamery. Tabulka 5 je doplněna grafem 3. Z hodnot uvedených v celkové tabulce (tabulka 2) vyplývá, že chybovala celkově v 12,19 procentech. Maximální počet chyb způsobený NXT kamerou byl 5 v jednom pokusu. Nejčastěji NXT kamera chybovala 3x v průběhu pokusů, a to u 5 pokusů z 20 mnou provedených. Myslím si, že chyby NXT kamery jsou způsobené šíří pásu, ale také tím, že pás je vroubkovaný a na třídící linku by byl vhodnější pás hladký. Obrázky chyb jsou uvedeny v kapitole 4.7.

číslo pokusů	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Počet chyb NXT kamery	2	5	4	3	3	3	0	3	1	0	3	0	0	0	1	1	2	0	0	0

Tabulka 5: Četnost chyb při detekci NXT kamery



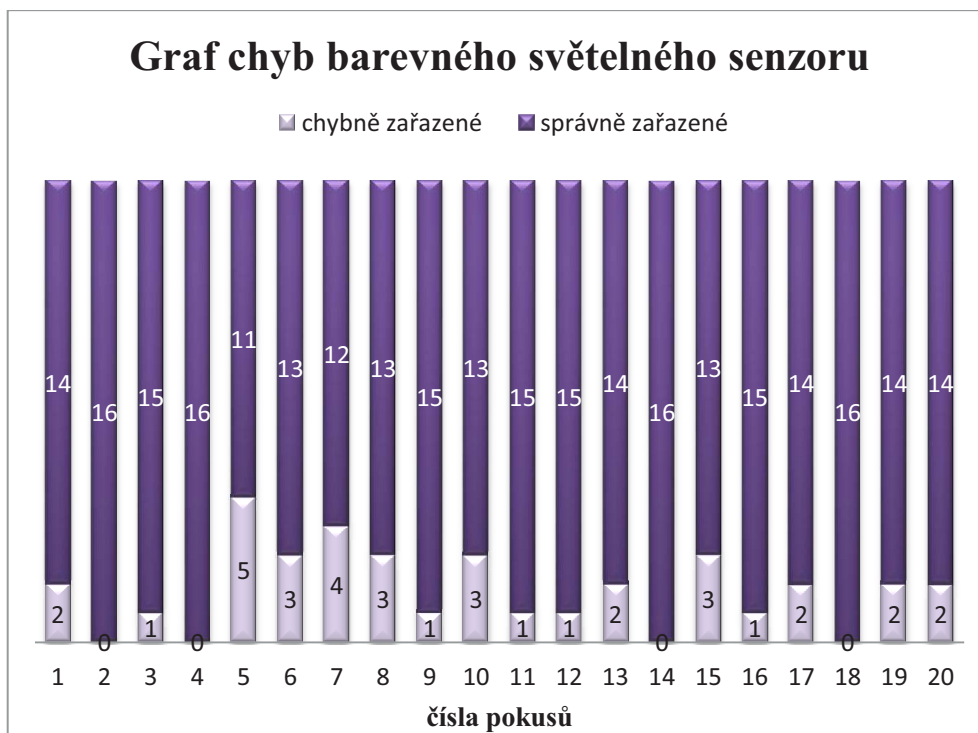
Graf 3: Zaznamenání chyb NXT kamery

4.4 Četnost chyb při detekci barevného světelného senzoru

V tabulce 6 je uveden počet chyb při detekci barevného světelného senzoru. Senzor slouží pro určení barvy tablet. Z provedených 20 pokusů byly pouze 4 pokusy bez chybného zařazení. Maximální počet chyb u barevného světelného senzoru bylo 5 a to u jednoho pokusu. Barevný světelný senzor udělal nejčastěji 1 nebo 2 chyby v 10 pokusech. Z hodnot zaznamenaných v celkové tabulce, (tabulce 2) můžeme vyčíst, že barevný světelný senzor chyboval celkově v 13,75 procentech. Tabulka 6 je doplněna grafem 4. Důvodem těchto chyb může být větší šířka posuvného pásu a malý rozsah záběru barevného světelného senzoru. Větší pravděpodobnost pro chybné zařazení mají menší tablety, protože mohou více sklouznout ke stranám, a tudíž nevyplní záběr barevného světelného senzoru a jsou chybně detekovány. Obrázky chyb jsou uvedeny v kapitole 4.6.

číslo pokusů	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Počet chyb barevného světelného senzoru	2	0	1	0	5	3	4	3	1	3	1	1	2	0	3	1	2	0	2	2

Tabulka 6: Četnost chyb při detekci barevného světelného senzoru



Graf 4: Zaznamenání chyb barevného světelného senzoru

4.5 Srovnání četnosti chyb u detekce NXT kamery a barevného světelného senzoru

Z hodnot v celkové tabulce (tabulka2) vyplývá, že NXT kamera při detekci celkově chybovala o 1,56 procent méně než barevný světelný senzor, což je o 5 tablet méně. Celkově jsem ve 20 pokusech zařazovala 320 tablet, z toho bylo 36 tablet zařazeno špatně podle barvy, 31 podle tvaru a velikosti a 8 tablet podle tvaru, velikosti i barvy. V grafu 5 je uvedeno srovnání chybně zařazených tablet u obou senzorů v jednotlivých pokusech.

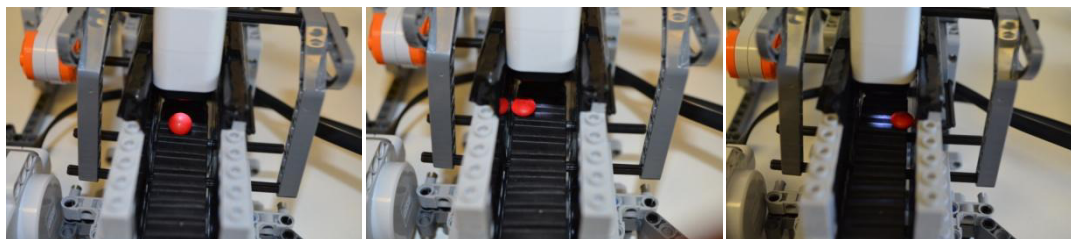


Graf 5: Srovnání četnosti chyb u detekce NXT kamery a barevného světelného senzoru

4.6 Zdůvodnění správného a špatného zařazení tablety u barevného světelného senzoru

Na obrázku 36 je vidět tableta, která projíždí pod barevným světelným senzorem a je správně detekována, nachází se ve středu pásu. Barevný světelný senzor může správně rozlišit barvu a tabletu správně zařadit.

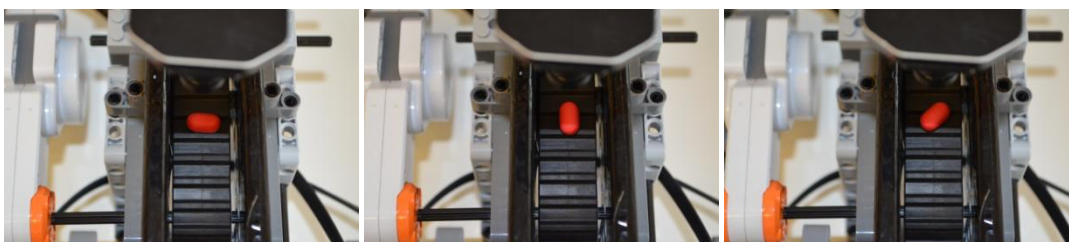
Na obrázcích 37 a 38 je možné vidět špatnou polohu tablety na posuvném pásu (tablety jsou u okraje posuvného pásu), tudíž detekce nemohla být provedena správně a tableta byla zařazena do zásobníku chybně. V případě, že by byl posuvný pás užší nebo disponoval vyrovnávacími klapkami, aby byla tableta přesně pod barevným světelným senzorem, byla by zařazena správně. Další možnou variantou je senzor s větším rozsahem záběru LED diodového světla.



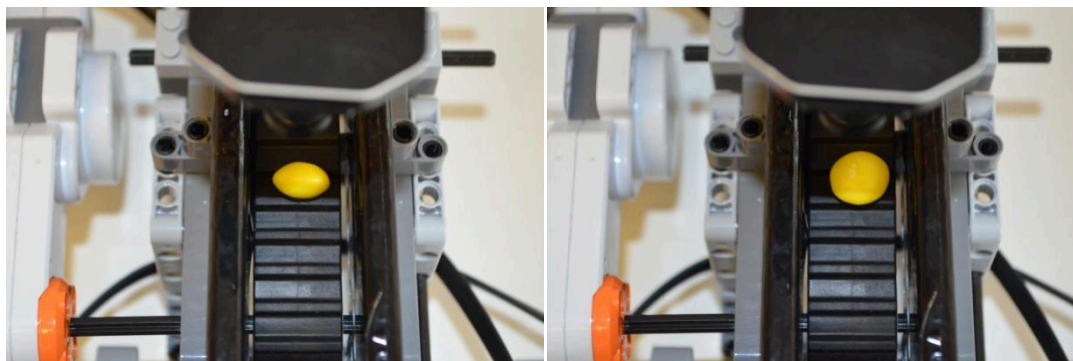
Obrázek 36: Správná poloha Obrázek 37: Chybná poloha Obrázek 38: Chybná poloha

4.7 Zdůvodnění správného a špatného zařazení tablety u NXT kamery

Na obrázku 39, 40 a 43 je vidět tableta, která projíždí pod NXT kamerou a je správně detekována. NXT kamera může správně rozlišit tvar a velikost tablety a zařadit do zásobníku. Na obrázku 41 a 42 můžete vidět špatnou polohu tablety na posuvném pásu. Detekce neproběhne správně. U kulatých tablet může dojít k umístění na posuvný pás zešíkma z důvodů vroubkovaného pásu a tudíž není tvar správně určen. Dalším důvodem může být malé rozlišení NXT kamery, které je pouze 88x144px.



Obrázek 39: Správná poloha Obrázek 40: Správná poloha Obrázek 41: Chybná poloha



Obrázek 42: Chybná poloha

Obrázek 43: Správná poloha

Závěr

Cílem této bakalářské práce bylo sestavit a vytvořit algoritmus pro chod mnou navržené třídící linky pro automatizovanou detekci a třídění typu tablet pomocí obrazového signálu. Třídící linku jsem sestavila pomocí stavebnice LEGO MINDSTORMS. Algoritmus jsem vytvořila v programu RobotC for MINDSTORMS 2.02. Tablety byly tříděny dle velikosti, tvaru a barvy. Měla jsem k dispozici tablety o dvou velikostech, a to malé a velké, tvar byl kulatý a podlouhlý, ty byly detekovány pomocí NXT kamery. Tři zvolené barvy byly červená, žlutá a růžová a jejich detekci zajišťoval barevný světelný senzor. Pro můj pokus jsem místo tablet používala tuhé cukrovinky typu lentilky atd.

Provedla jsem 20 pokusů, přičemž jeden pokus detekoval 16 tablet. Výsledky pokusů jsem shrnula v celkové tabulce (tabulka 2). Ze získaných výsledků je možné vyvodit, že barevný světelný senzor měl větší množství chyb než NXT kamera, ale jen o 5 kusů tablet. Celková úspěšnost správně zařazených tablet je 76,56 %. U dvou sérií pokusů proběhla detekce tablet se stoprocentně správnou úspěšností. V 7 pokusech se vyskytla chyba současně u barevného světelného senzoru i kamery. NXT kamera celkově chybovala v 12,19 procentech, tj. 39 kusů tablet bylo špatně přiřazeno dle velikosti nebo tvaru. Barevný světelný senzor celkově chybně zařadil v 13,75 procentech, tj. 44 tablet bylo zařazeno k jiné barvě.

Možnou příčinou chyb mohl být zvolený typ posuvného pásu a jeho šířka. Dalším důvodem vyššího počtu chyb může být úzký záběr barevného světelného senzoru. NXT kamera použita k pokusům měla malé rozlišení (88x144px). Výsledný obraz u kamery může ovlivnit nasvícení bílou zářivkou. Ve mnou navržené třídící lince jsem pracovala jen s přirozenými světelnými podmínkami.

Seznam použité literatury

- [1] CHALABALA, Milan. *Technologie léků*. 2. přeprac. a dopl. vyd. Praha: Galén, 2001, 408 s. ISBN 80-726-2128-9.
- [2] NEUWIRTH, J. *Klinická propedeutika*. 1988. vyd. Praha: Avicenum, 1988.
- [3] DOBEŠ, Michal. *Zpracování obrazu a algoritmy v C#*. 1.vyd.Praha:BEN-technická literatura, 2008, 144s. ISBN 978-80-7300-233-6.
- [4]HLAVÁČ,Václav. SEDLÁČEK, Miloš. *Zpracování signálů a obrazu*.vyd.ČVUT, 2001, 220s. ISBN 80-01-02114-9
- [5] KADLEC, Václav. *Učíme se programovat v jazyce C*. vydání druhé. vyd. CP Books, 2005,402s. ISBN 80-7226-715-9

Internetové zdroje

- [6] LÉČIVO, LÉČIVÉ PŘÍPRAVKY, LÉKY, LÉKOVÉ FORMY. HAUSEROÁ, Dana. [online]. [cit. 2013-01-18]. Dostupné z: http://www.szs-jaselska.cz/_doc/projekty/kla_1_metodicky_list_formy_leku.ppt
- [7] Encyklopedie SUKL. [online]. [cit. 2013-01-18]. Dostupné z: <http://www.olecich.cz/encyklopedie/co-je-to-lekova-forma-a-jake-druhy-jsou>
- [8] JAKEŠ, T. *Robotické vzdělávání* [online]. 2010. vyd. [cit. 2013-01-18]. Dostupné z: <https://lego.zcu.cz/web/>
- [9] LEGO. *LEGO* [online]. [cit. 2013-01-19]. Dostupné z: <http://www.lego.com/cs-cz/>
- [10] Lékové formy. [online]. [cit. 2014-04-29]. Dostupné z: <tresen.vscht.cz/min/sites/default/files/lfpc2.ppt>
- [11] RobotC. *RobotC* [online]. [cit. 2014-04-29]. Dostupné z: <http://www.robotc.net/download/>
- [12] Mindsensors. [online]. [cit. 2014-03-30]. Dostupné z: http://www.mindsensors.com/index.php?module=pagemaster&PAGE_user_op=view_page&PAGE_id=78

Bakalářská práce

- [13] TROJÁNEK, Pavel. *Využití robota LEGO MINDSTORMS při výuce*. Praha 2009. Bakalářská práce. (Bc.). České vysoké učení technické v Praze. Fakulta elektrotechnická

Seznam obrázků

Obrázek 1: Paralen	3
Obrázek 2: Dologran	3
Obrázek 3: Celaskon	4
Obrázek 4: Geriavit	4
Obrázek 5: Wobenzym.....	4
Obrázek 6: Dobexil H	5
Obrázek 7: Tvary tablet.....	6
Obrázek 8: Ukázka použitých cukrovinek pro simulaci léčiv	6
Obrázek 9: LEGO	7
Obrázek 10: Základní připojení vstupních a výstupních zařízení k portům řídicí jednotky NXT.....	8
Obrázek 11: NXT kostka Obrázek 12: Vnitřní propojení modulu	9
Obrázek 13: Spojovací kabel NXT Obrázek 14: Rozdíl koncovek.....	10
Obrázek 15: Světelný senzor.....	11
Obrázek 16: Dotykový senzor.....	11
Obrázek 17: Barevný světelný senzor	12
Obrázek 18: Interaktivní servomotor Obrázek 19: Vnitřní uspořádání interaktivního servomotoru	12
Obrázek 20: NXT kamera	13
Obrázek 21: Model RGB.....	15
Obrázek 22: Model CMY.....	15
Obrázek 23: Ukázka rozhraní v programu RobotC.....	17
Obrázek 24: Aktualizace firmware	18
Obrázek 25: Nahrání firmware.....	18
Obrázek 26: úvodní stránka NXTCamView	19
Obrázek 27: NXTCamView připravený k načtení objektu	20
Obrázek 28: načtený objekt v programu NXTCamView	20
Obrázek 29: Návrh řešení třídící linky.....	21
Obrázek 30: Realizace řešení třídící linky	22
Obrázek 31: Princip algoritmu třídící linky	22
Obrázek 32: Vývojový diagram třídící linky	23
Obrázek 33: Kalibrované barvy	25
Obrázek 34: Barevný světelný senzor	25
Obrázek 35: Ukázka načtení objektu v programu Camview.....	27
Obrázek 36: Správná poloha Obrázek 37: Chybná poloha Obrázek 38: Chybná	35
Obrázek 39: Správná poloha Obrázek 40: Správná poloha Obrázek 41: Chybná.....	36
Obrázek 42: Chybná poloha Obrázek 43: Správná poloha	36

Seznam tabulek

Tabulka 1: Přehled možností programování robota	16
Tabulka 2: Naměřené hodnoty	29
Tabulka 3: Chybně a správně zařazené hodnoty v %.....	30
Tabulka 4: Četnost chyb při detekci obou senzorů	31
Tabulka 5: Četnost chyb pro detekci NXT kamery.....	32
Tabulka 6: Četnost chyb při detekci barevného světelného senzoru.....	33

Seznam grafů

Graf 1: Graf úspěšnosti pro správně a špatně zařazené tablety	30
Graf 2: Současné zaznamenání chyb NXT kamery i barevného světelného senzoru	31
Graf 3: Zaznamenání chyb NXT kamery	32
Graf 4: Zaznamenání chyb barevného světelného senzoru	33
Graf 5: Srovnání četnosti chyb u detekce NXT kamery a barevného světelného senzoru.....	34

Seznam použitých programů

RobotC for MINDSTORMS 2.02

NXTcamview

Microsoft Office Word

Microsoft Office Excel

Dia

Seznam příloh

Příloha 1: Zdrojový kód

Příloha 2: Fotodokumentace třídící linky

Příloha 3: Obsah CD

Příloha 1: Zdrojový kód

```

/*****
/**AUTOMATIZOVANA DETEKCE A TRIDENI TYPU TABLET POMOCI OBRAZOVEHO SIGNALU**/
**/
**/
VEC0018_BP.C
**/
**/
Ivana Večerková
**/
Ostrava 2014
**/
*****/

#pragma config(Sensor,S1,HTCS,sensorI2CCustomStd)

// Importuje základní funkce pro komunikaci s NXT kostkou
#include "common.h"
// Importuje funkce pro práci s kamerou
#include "nxtcamlib.c"
// Importuje funkce pro práci s barevným světelným senzorem
#include "HTCS-driver.h"

const tSensors cam = (tSensors) S2;
// Pomocne konstanty barev
const int CERVENA = 8;
const int ZLUTA1 = 5;
const int ZLUTA2 = 6;
const int RUZOVA = 10;

// Vytvorime si promennou, ve které si budeme ukladat nacteny tvar
// Necht 0 = ctverec, 1 = obdelnik
int tvar = 0;
int color;
int last_color = 0;
int velikost = 0; // 0 = mala, 1 = velka

// Pomocna funkce pro skalovani x
int xscale(int x) {
    return ((x - 12) * 99) / 176;
}

// Pomocna funkce pro skalovani y
int yscale(int y) {
    return ((143 - y) * 63) / 143;
}

// Funkce spusti motory pro pas
// Motory pasu musi byt v socketu A a B
void motor_pas_on() {
    motor[motorA] = -5;
    motor[motorB] = -5;
}

// Funkce zastavi motory pro pas
void motor_pas_off() {
    motor[motorA] = 0;
    motor[motorB] = 0;
}

// Pokud je nactena velikost mala,
// presunout motor na pozici "mala"
void motor_velikost_mala() {
    motor[motorC] = 17;
}

```

```
        wait1Msec(2200);
motor[motorC] = 0;

    velikost = 0;
}

// Pokud je nactena velikost velka,
// presunout motor na pozici "velka"
void motor_velikost_velka() {
    motor[motorC] = -17;
    wait1Msec(2200);
    motor[motorC] = 0;
    velikost = 1;
}

// Funkce se spusti pokud kamera nacetla ctverec
void motor_ctverec(float v) {
    // Pokud uz predchozi tvar byl ctverec, nebude se nic delat
    // pas bude na svem miste. Pokud predchozi tvar nebyl ctverec
    // vykona se podminka - motor se spusti na 4,3s a tam dojede
    // na spravne misto na pasu
    if(tvar!=0) {
        motor[motorC] = 17;
        wait1Msec(4300);
        motor[motorC] = 0;
    }
    tvar = 0;
    // Pokud je velikost ctverce mala a predchozi
    // velikost ctverce byla velka, presun pas na pozici
    // mala
    if(v<=1000&&velikost==1) {
        motor_velikost_mala();
    }

    // Pokud je velikost ctverce velka a predchozi
    // velikost byla mala, presun pas na pozici
    // velka
    if(v>1000&&velikost==0) {
        motor_velikost_velka();
    }
}

// Funkce se spusti pokud kamera nacetla obdelnik
void motor_obdelnik(float v) {
    // Pokud predchozi tvar byl ctverec, vykona se tato podminka
    // Pokud byl obdelnik, neni co resit - viz vyse.
    if(tvar==0) {
        motor[motorC] = -17;
        wait1Msec(4300);
        motor[motorC] = 0;
    }
    tvar = 1;
    // Pokud je velikost obdelniku mala a predchozi
    // velikost obdelniku byla velka, presun pas na pozici
    // mala
    if(v<=1000&&velikost==1) {
```

```

        motor_velikost_mala();
    }

    // Pokud je velikost obdelniku velka a predchozi
    // velikost byla mala, presun pas na pozici
    // velka
    if(v>1000&&velikost==0) {
        motor_velikost_velka();
    }
}

// Pokud senzor nacetl barvu zlutou a predchozi barva
// byla ruzova nebo cervena, presun pas na pozici "zluta"
void motor_zluta() {
    if(last_color==CERVENA||last_color==RUZOVA) {
        motor[motorC] = 17;
        wait1Msec(1100);
        motor[motorC] = 0;
    }
}

// Pokud senzor nacetl barvu červenou a predchozi barva
// byla zluta, presun pas na pozici "cervena"
void motor_cervena() {
    if(last_color==0||last_color==ZLUTA1||last_color==ZLUTA2)
    {
        motor[motorC] = -17;
        wait1Msec(1100);
        motor[motorC] = 0;
    }
}

task main() {
    int n; // Pocet objektu, ktere nacetla kamera
    int i;
    int_array bc, bl, bt, br, bb; // Pole krajnich hodnot tvorici
    tvar ctvercu
    int l, t, r, b; // Promenna uchovavajici krajni hodnoty
    jednotlivych ctvercu

    // Pomocna promenna, ktera se prepise podle toho, jestli kamera
    nasnimala
    // nejaky objekt nebo ne. Kdyz kamera nacte nejaky ctverec, zmeni se
    promenna
    // na false. Kdyz kamera nenacte zadny objekt, prepne se zpatky na
    true.
    // Pokud je promenna true, znamena to, ze kamera muze nacist dalsi
    objekt
    bool zmena_tvaru = true;
    bool zmena_barvy = true;

    // Inicializace kamery
    init_camera(cam);
    // Zapnout hlavni pas
    motor_pas_on();

    // Hlavni smycka
    while(true) {
        get_blobs(cam, n, bc, bl, bt, br, bb);
    }
}

```

```

// Vymazani displaye
eraseDisplay();

// pokud se zmenila barva, nacti ji
if(zmena_barvy) {
    color = HTCSreadColor(HTCS);
    nxtDisplayTextLine(0, "%d", color);
if(color!=12) {

    // Pokud je nactena barva cervena nebo ruzova
    // zavolej funkci motor_cervena()
    if(color==CERVENA||color==RUZOVA) {
        motor_cervena();
        zmena_barvy = false;
        last_color = color;
    }

    // Pokud je nactena barva zluta,
    // zavolej funkci motor_zluta()
    if(color==ZLUTA1||color==ZLUTA2) {
        motor_zluta();
        zmena_barvy = false;
        last_color = color;
    }
}
}

// Projde v cyklu vsechny objekty a vykresli je
for (i = 0; i < n; i++) {
    l = xscale(bl[i]); // kde zacina vykresleni x
    t = yscale(bt[i]); // kde zacina vykresleni y
    r = xscale(br[i]); // sirka
    b = yscale(bb[i]); // vyska
    float delta_x = r-l;
    float delta_y = t-b;
    float obsah;

    // Filtr. Filtruje sum. Pokud kamera zachyti nejakou drobnost
    // která ma obsah mensi nez 25px, tak bude ignorovat
    if(zmena_tvaru&&(delta_x*delta_y<25)) continue;
    nxtFillRect(l, t, r, b);

    // Pokud predtim bylo pred kamerou prazdno, tak se pokusi
    nastavit dalsi objekt
    if(zmena_tvaru) {

        // Pokud je objekt za urcitou hranici pred kamerou (32),
        nacti jeho tvar
        if(t>32) {
            obsah = delta_x*delta_y;

            // Pokud je jedna strana vice-mene stejne dlouha jako
            druha, tak se jedna o ctverec
            // a spusti se funkce motor_ctverec();
            if((delta_x>delta_y&&(delta_x<delta_y+delta_x/3.5))|
            |(delta_y>delta
                nxtDisplayTextLine(0, "kulata");

```

```
        zmena_barvy    = true;
        motor_ctverec(obsah);
    }

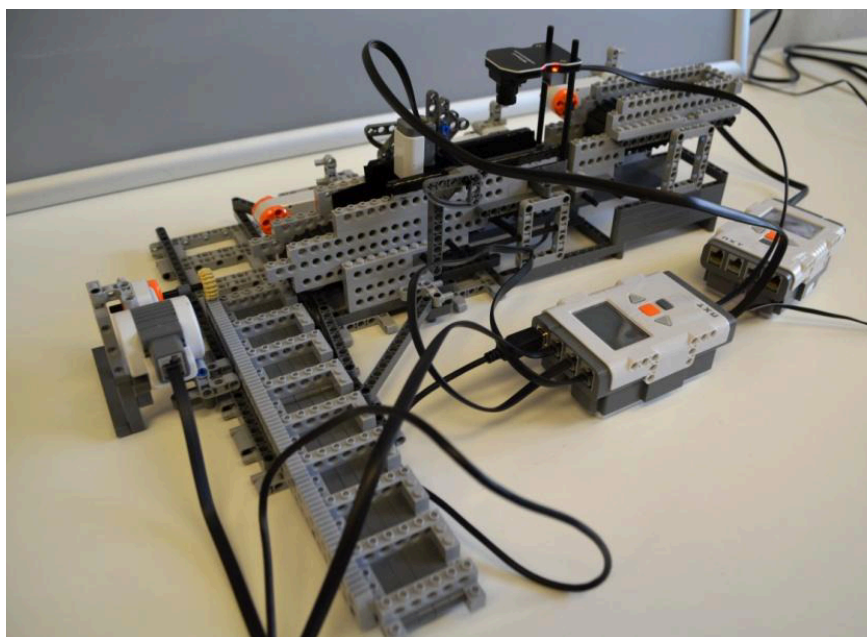
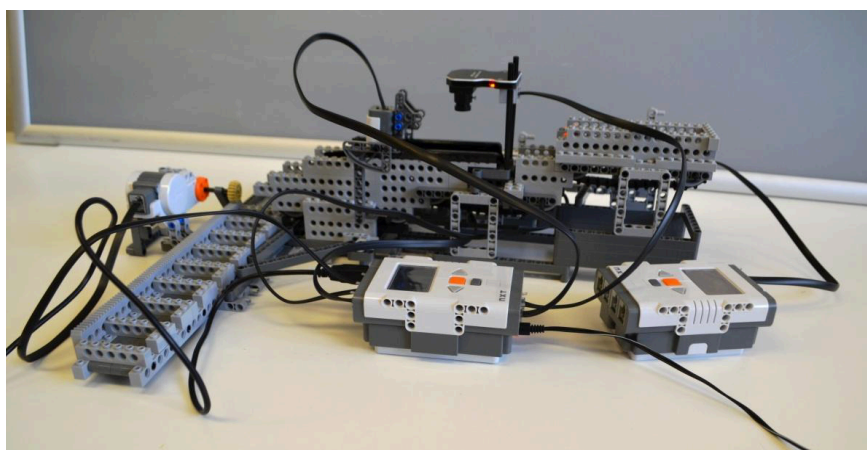
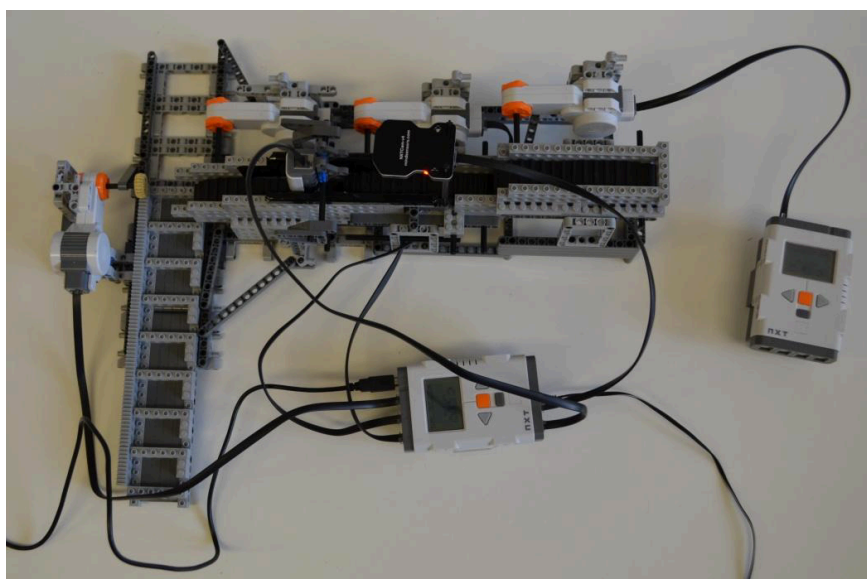
    // Pokud je jedna strana o nieco delsi, tak je to obdelnik
    else {
        nxtDisplayTextLine(0,"podlouhla");
        zmena_barvy    = true;
        motor_obdelnik(obsah);
    }
    zmena_tvaru = false;
}

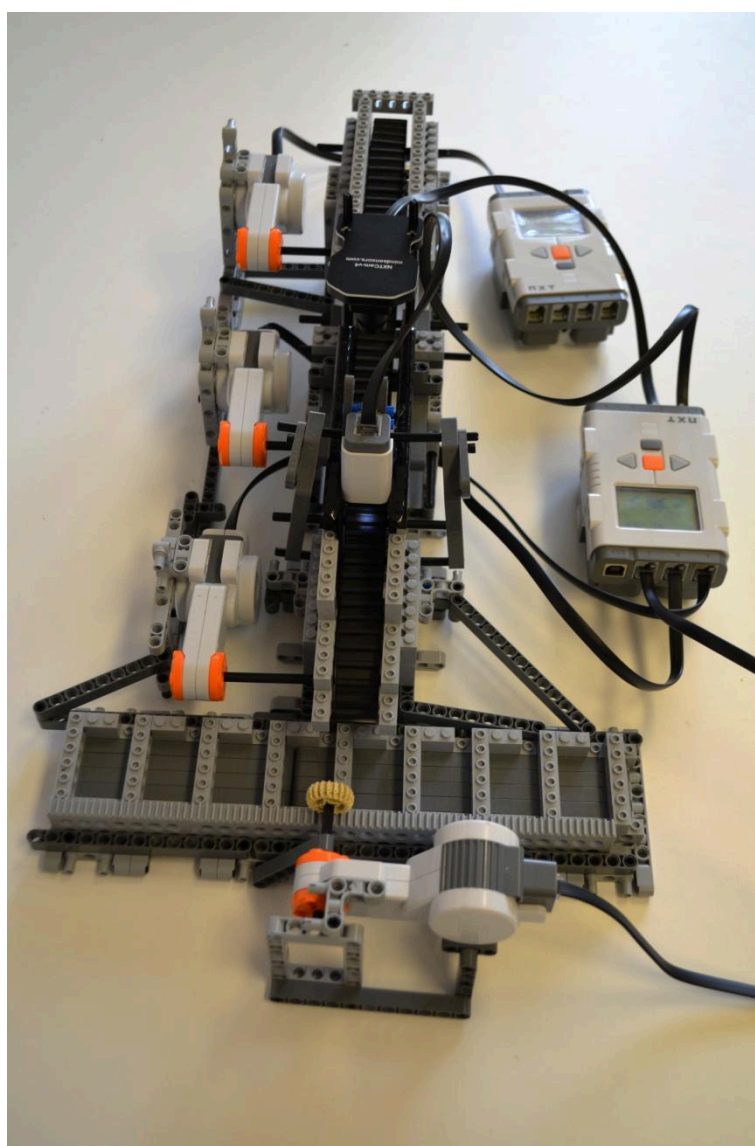
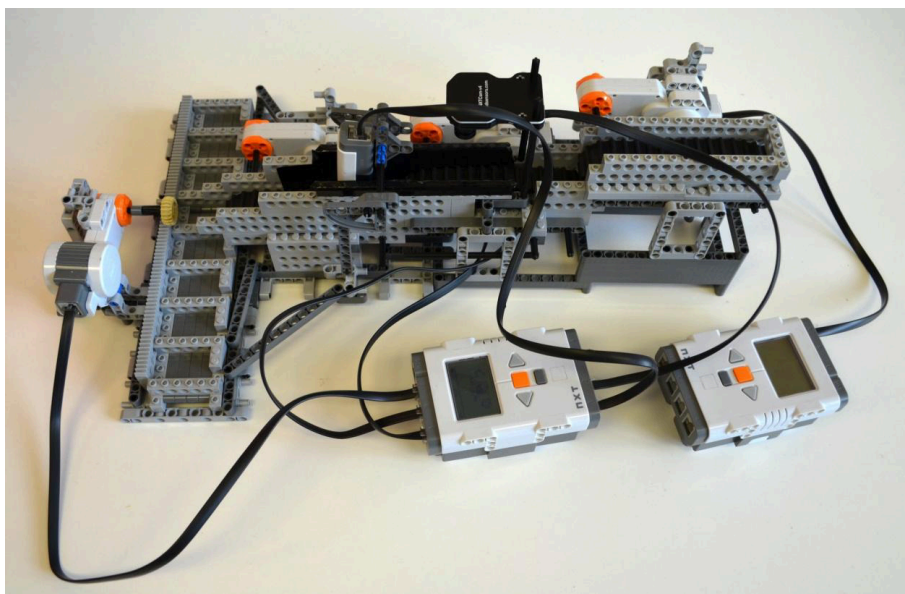
// Pokud pred kamerou momentalne nic neni prepne se promenna
// zmena_tvaru do stavu true
// a tam se rekne, ze se ma pokusit nastavit dalsi objekt

else {
    // t - kontrola strany - pokud je 0, pred kamerou nic neni
    if(t==0) {
        zmena_tvaru = true;
    }
}

}
```


Příloha 2: Fotodokumentace třídící linky





Příloha 3: Obsah CD

Bakalářská práce

Zdrojový kód

Fotodokumentace třídící linky

Video třídící linky